



universität  
wien

# DIPLOMARBEIT

Titel der Diplomarbeit

Energie-Casimir-Normen zur Störungsgenerierung in  
Ensemblevorhersagen

Verfasserin

Johanna Bugkel

Angestrebter akademischer Grad

Magistra der Naturwissenschaften  
(Mag. rer. nat.)

Wien, 2012

Studienkennzahl: A 415

Studienrichtung: Diplomstudium Meteorologie

Betreuer: Ao. Univ.-Prof. Mag. Dr. Leopold Haimberger

*“La gravitation de l’esprit nous fait  
tomber vers le haut.”*

*„Die Schwerkraft des Geistes läßt  
uns nach oben fallen“.*

Simone Weil (1909-43)

# Inhaltsverzeichnis

<b>Zusammenfassung</b>	<b>vi</b>
<b>Abstract</b>	<b>vii</b>
<b>Teil I</b>	<b>1</b>
<b>1 Einleitung</b>	<b>2</b>
<b>2 Das Lorenz(1963)-Modell in 3 Dimensionen</b>	<b>4</b>
2.1 Herleitung . . . . .	4
2.1.1 Grundgleichungen . . . . .	4
2.1.2 HENRI BÉNARD - Das Experiment . . . . .	5
2.1.3 LORD RAYLEIGH - Die Konvektionsgleichungen . . . . .	8
2.1.4 BARRY SALTZMAN - Die Integration . . . . .	12
2.1.5 EDWARD LORENZ - Die Lorenz-Gleichungen . . . . .	14
2.1.6 Alternative Herleitung . . . . .	14
2.2 Eigenschaften . . . . .	19
<b>3 Alternative Formulierungen der klassischen Mechanik</b>	<b>31</b>
3.1 Mathematische Modelle der klassischen Mechanik . . . . .	32
3.1.1 Mechanik nach Newton . . . . .	32
3.1.2 Lagrange-Mechanik . . . . .	32
3.1.3 Hamilton-Mechanik . . . . .	35
3.1.4 Nambu-Mechanik . . . . .	41
3.2 Betrachtung des Lorenz(1963)-Modells in Hamilton- bzw. Nambu-Mechanik	42
3.2.1 Hamiltonsche Formulierung der Lorenz-Gleichungen . . . . .	42
3.2.2 Nambu-mechanische Formulierung der Lorenz- Gleichungen . . . .	44
3.2.3 Problematik der Lorenz-Gleichungen . . . . .	47
<b>4 Erweiterung des Lorenz(1963)-Modells auf 6 Dimensionen</b>	<b>50</b>
4.1 Herleitung . . . . .	50

<b>5</b>	<b>Anwendung und Implementierung</b>	<b>59</b>
5.1	Programmaufbau im Überblick . . . . .	60
5.2	Der Programmcode und theoretische Hintergründe . . . . .	62
5.2.1	rbc_6dim_autom.m . . . . .	62
5.2.2	rbc_6dim.m . . . . .	65
5.2.2.1	Modellparameter . . . . .	65
5.2.2.2	Variablendefinition . . . . .	66
5.2.2.3	Berechnung . . . . .	67
5.2.2.4	Verifikation . . . . .	89
5.2.2.5	Plots . . . . .	92
5.2.2.6	Functions . . . . .	92
<b>6</b>	<b>Ergebnisse</b>	<b>94</b>
6.1	Lyapunov Analyse . . . . .	95
6.2	Phasenraumplots . . . . .	95
6.3	Zeitreihe der Komponenten . . . . .	101
6.4	Temperatur- und Vorticityfelder . . . . .	104
6.5	Stabilität . . . . .	106
6.6	Größe der Störungen . . . . .	112
6.7	Bewertung des EPS - Optimierung durch den Energie-Vorticity-Casimir .	113
6.7.1	Brier Score . . . . .	113
6.7.2	Reliability-Diagramm, Resolution-Diagramm . . . . .	127
6.7.3	ROC-Diagramm . . . . .	129
6.8	Verwendung anderer Normkonstruktionen . . . . .	130
6.8.1	Energie-Temperatur-Casimir . . . . .	130
6.8.2	Energie-Helizitäts-Temperatur-Casimir . . . . .	136
<b>7</b>	<b>Conclusio und Ausblick</b>	<b>138</b>
<b>Teil II:</b>	<b>Anhang</b>	<b>143</b>
<b>Abbildungen</b>		<b>144</b>
<b>Txt-Outputs</b>		<b>155</b>
<b>Matlab-Codes</b>		<b>160</b>



# Variablenverzeichnis

$\vec{g}$  Gravitationsvektor in  $[\text{m/s}^2]$

$\alpha$  Wärmeausdehnungskoeffizient in  $[1/\text{K}]$

$\eta$  dynamische Viskosität in  $[\text{Ns/m}^2]$

$\kappa$  Temperaturleitfähigkeit in  $[\text{m}^2/\text{s}]$

$\lambda$  Wärmeleitfähigkeit in  $[\text{W}/(\text{mK})]$

$\nu$  kinematische Viskosität in  $[\text{m}^2/\text{s}]$

$\rho$  Dichte in  $[\text{kg/m}^3]$

$\sigma$  Prandtl-Zahl

$\Theta$  Abweichung vom linearen Temperaturprofil in  $[\text{K}]$

$\vec{v}$  Geschwindigkeitsvektor je in  $[\text{m/s}]$

$\vec{x}$  Ortsvektor in  $[\text{m}]$

$c_p$  spezifische Wärmekapazität bei konstantem Druck in  $[\text{J}/(\text{kgK})]$

$R_a$  Rayleigh-Zahl

$p$  Druck in  $[\text{N/m}^2]$

$r$  relative Rayleigh-Zahl

$T$  Temperatur in  $[\text{K}]$

# Zusammenfassung

Untersucht werden in dieser Arbeit Verbesserungsmöglichkeiten in der Ensemblevorhersage anhand einer 6-dimensionalen Erweiterung des Lorenz(1963)-Modells.

Dem soll jedoch eine detaillierte Beschreibung des Weges, ausgehend von den atmosphärischen Grundgleichungen bis hin zum Konzept des *finite-mode models* vorangestellt werden.

Aus den drei Erhaltungssätzen von Impuls, Masse und Energie werden zunächst die Gleichungen der Rayleigh-Bénard-Konvektion hergeleitet, diese werden dann mit Hilfe numerischer Untersuchungen auf die bekannten Lorenz-Gleichungen reduziert.

Nach einer Einführung in die alternativen Formulierungen der klassischen Mechanik soll unter Verwendung des Hamiltonschen Formalismus gezeigt werden, warum das originale Lorenz(1963)-Modell nicht als *maximum simplification* der Konvektionsgleichungen angesehen werden kann. Unter Verwendung der Nambu-Mechanik wird die Herleitung des auf 6 Dimensionen erweiterten Lorenz-Modells skizziert.

Dieses Modell, genauer gesagt dessen konservativer Kern, wird anschließend implementiert und eine Ensemblevorhersage gestartet.

Für ein EPS (Ensemble Prediction System) sehr wichtig sind die generierten Störungen, die nach ihrer Entwicklung zeigen, in welchem Bereich sich der Endzustand unter Berücksichtigung einer Unsicherheit im Anfangszustand befinden kann.

Zur Generierung dieser Störungen werden oftmals reine Energienormen verwendet, das entspricht der Suche nach speziell den Störungen, die einen maximalen Fehler in der Gesamtenergie des Endzustandes bewirken.

Da es in der realen Atmosphäre neben der Erhaltung der Gesamtenergie auch noch die Erhaltung der Ertelschen potentiellen Vorticity gibt, existieren bereits theoretische Überlegungen, die es nahelegen, hier ebenfalls eine kombinierte *Energie-Vorticity-Norm* zur Ermittlung der optimalen Störungen zu verwenden.

Diese Einführung neuer Normen wird im programmierten Modell realisiert und deren Auswirkung anhand diverser Verifikationsmaße untersucht.

# Abstract

This thesis investigates the possibility of improving ensemble prediction using a 6-dimensional expansion of the Lorenz(1963)-model.

Prior to this, a detailed description of the derivation, from the primitive equations that govern atmospheric motions to the *finite-mode model* concept is presented.

From the laws of conservation of momentum, mass and energy, the equations of Rayleigh-Bénard-convection are derived and subsequently reduced to the well-known Lorenz equations by means of numerical considerations.

After an introduction to the alternative formulations of classical mechanics, the reason why the original Lorenz(1963)-model cannot serve as *maximum simplification* of the convection equations is shown, using the Hamiltonian mechanics.

Nambu mechanics are then employed to outline the derivation of the Lorenz model expanded to 6 dimensions.

This model (or rather its conservative core) is subsequently implemented and an ensemble prediction initiated. The generated perturbations that indicate upon their formation the possible range for the final state, given an uncertainty in the initial state, are of great importance to an EPS (Ensemble Prediction System).

Currently, a pure energy norm is often used in order to generate these perturbations. This corresponds to the search of perturbations that will specifically cause a maximum error in total energy of the final state.

Since the real atmosphere is governed by conservation of Ertel potential vorticity in addition to total energy conservation, theoretical considerations have already been presented which suggest the use of a combined energy-vorticity-norm in determining optimal perturbations for this case as well.

The introduction of such novel norms is realised within the programmed model, and its effect is studied by means of various verification standards.

# Teil I

# 1 Einleitung

Von großer Bedeutung für die Dynamik der Atmosphäre ist die treibende Kraft der *Konvektion*, die zumeist durch inhomogene Temperaturverteilungen zustande kommt. Sie reicht von der Skala der globalen Zirkulation aufgrund der unterschiedlichen solaren Einstrahlung über lokale Entwicklungen von Zyklonen bis hin zu turbulenten kleinskaligen Phänomenen.

Jedoch haben all diese unterschiedlichen Konvektionsmechanismen eines gemeinsam: den gleichzeitigen Transport von Wärme und Vorticity, welcher zu einem großteils nichtlinearen Verhalten atmosphärischer Strömungen führt. [32]

Zwar lassen sich Gleichungen finden, die bestimmte Mechanismen in guter Näherung beschreiben, allerdings handelt es sich hierbei oft, wie auch bei den Rayleigh-Bénard-Konvektionsgleichungen, um nichtlineare partielle Differentialgleichungssysteme mit unendlich vielen Freiheitsgraden, deren direkte Betrachtung relativ wenig Einblick in den zugrunde liegenden physikalischen Prozess gewährt.

Dies ist ein Grund für die Suche nach Modellvereinfachungen mit dem Ziel, eine maximale Reduktion der Freiheitsgrade unter Beibehaltung der ausschlaggebenden Charakteristika der Originalgleichungen zu gewährleisten. Weiters rechtfertigen auch deutliche Einsparung der Rechenleistung zur Prognose des atmosphärischen Zustands den hohen Aufwand der Entwicklung solcher Modelle.

Die unendlich-dimensionalen partiellen Differentialgleichungen werden zu endlich-dimensionalen gewöhnlichen Differentialgleichungssystemen reduziert. Diese Art von Modellen wird als *maximum simplification*, *low-order model*, *finite -mode model* usw. bezeichnet.

Es stellt sich die wichtige Frage: Welche Moden müssen nun beibehalten werden um einer ausreichenden Beschreibung der Systemdynamik gerecht zu werden?

Zu diesem Thema gibt es bereits zahlreiche wissenschaftliche Arbeiten, hier vorgestellt werden das 3-dimensionale Lorenz(1963)-Modell und dessen Erweiterung auf 6 Dimensionen [7], welche beide Reduktionen der Rayleigh-Bénard-Konvektionsgleichungen darstellen. Es soll auch die Notwendigkeit dieser Erweiterung dargelegt werden und gezeigt werden, dass das ursprüngliche Modell von 1963 nach neusten Erkenntnissen eigentlich nicht als *maximum simplification* angesehen werden kann.

## 1 Einleitung

Nicht nur der physikalische Mechanismus der Konvektion selbst, sondern auch die mathematische Behandlung dieser Gleichungssysteme, ausgezeichnet durch das *deterministisches Chaos*, soll im Vordergrund stehen.

Diese Modelle siedeln sich mathematisch gesehen im Gebiet der Chaostheorie an, deren Definition nach [41] einen kurzen Vorgriff zur Beschreibung der Systemeigenschaften geben soll:

“*Chaostheorie* beschäftigt sich mit Systemen, denen zwar deterministische Gesetzmäßigkeiten zugrunde liegen, deren Verhalten jedoch irregulär und langfristig nicht vorhersagbar ist.

- Die zugrunde liegende DGLs sind nicht linear ( Nichtlineare Dynamik).
- Folge: kleine Variation in den Anfangsbedingungen verstärken sich exponentiell.
- Vorhersage über genauen Systemzustand nur begrenzt berechenbar bzw. unmöglich.”

Weil solche Modelle nur wenig Rechenleistung in Anspruch nehmen, eignen sie sich hervorragend für Experimente bezüglich Ensemblevorhersagen. Diesen Vorteil nutzend soll in der Anwendung des erweiterten 6-dimensionalen Modells nach Verbesserungsmöglichkeiten in der Ensemblevorhersage gesucht werden, die auf der Verwendung von Erhaltungsgrößen basieren, welche im originalen 3-dimensionalen Modell nicht existieren.

## 2 Das Lorenz(1963)-Modell in 3 Dimensionen

### 2.1 Herleitung

Um ausgehend von den fundamentalen physikalischen Erhaltungssätzen Impuls-, Massen- und Energieerhaltung zum berühmten *Lorenz(1963)-Modell* zu gelangen, müssen die geistigen Errungenschaften verschiedener Mathematiker und Physiker über mehrere Jahrzehnte betrachtet werden.

- HENRI BÉNARD - führte Experimente zum Thema Konvektion durch.
- LORD RAYLEIGH - beschäftigte sich mit der theoretischen Beschreibung und Verfassung physikalischer Gesetze für Bénards Versuche
- BARRY SALTZMAN - wählte für die sogenannten Rayleigh-Bénard-Gleichungen konkrete Randbedingungen, entwickelte die dynamischen Variablen in doppelte Fourierreihen und löste das Gleichungssystem numerisch.
- EDWARD LORENZ - verwendete die Ergebnisse der numerischen Lösung von Saltzman um ein reduziertes Gleichungssystem zu entwickeln - das 3-dimensionale Lorenz(1963)-Modell.

#### 2.1.1 Grundgleichungen

Um den Zustand der Atmosphäre mathematisch zu beschreiben, werden die Dichte  $\rho$ , der Druck  $p$ , die Temperatur  $T$  und die Strömungsgeschwindigkeit  $\vec{v}$  als kontinuierliche Feldgrößen aufgefasst:

$$\rho = \rho(\vec{x}, t)$$

$$p = p(\vec{x}, t)$$

$$T = T(\vec{x}, t)$$

$$\vec{v} = \vec{v}(\vec{x}, t) = \begin{pmatrix} u(\vec{x}, t) \\ v(\vec{x}, t) \\ w(\vec{x}, t) \end{pmatrix}$$

wobei

$$\vec{x} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

der Ortsvektor in kartesischen Koordinaten ist.

Für die Herleitung der Rayleigh-Bénard-Gleichungen werden folgende Grundgesetze benötigt:

Bewegungsgleichung - Impulserhaltung

$$\rho \frac{d\vec{v}}{dt} = \rho \left( \frac{\partial \vec{v}}{\partial t} + \vec{v} \cdot \nabla \vec{v} \right) = \rho \vec{g} - \nabla p + \eta \nabla^2 \vec{v} \quad (2.1)$$

Kontinuitätsgleichung - Massenerhaltung

$$\frac{1}{\rho} \frac{\partial \rho}{\partial t} + \nabla \cdot \vec{v} = 0 \quad (2.2)$$

Wärmetransportgleichung - Energieerhaltung

$$\rho c_p \frac{dT}{dt} = \rho c_p \left( \frac{\partial T}{\partial t} + \vec{v} \cdot \nabla T \right) = \lambda \Delta T \quad (2.3)$$

Dabei ist  $\vec{g}$  der Gravitationsvektor,  $c_p$  die spezifische Wärmekapazität bei konstantem Druck,  $\eta$  die dynamische Viskosität und  $\lambda$  die Wärmeleitfähigkeit.[18]

## 2.1.2 HENRI BÉNARD - Das Experiment

Literatur: [18, 19, 34, 37]

Zu den grundlegendsten Erkenntnissen der Thermodynamik zählt das Streben jedes Systems nach seinem thermischen Gleichgewicht, das durch verschiedene Mechanismen realisiert werden kann.

Neben dem konduktiven Wärmetransport, der pro Volumen stets proportional zu dem



vorliegenden Temperaturgradienten ist, gibt es in Fluiden zusätzlich die weitaus effektivere Wärmeübertragung durch Konvektion. Hierbei entwickelt sich ein Wärmefluss entgegen das Gravitationsfeld durch die geringere Dichte eines relativ zu seiner Umgebung wärmeren Fluids.

Einschlägige physikalisch-experimentelle Beiträge lieferte auf diesem Forschungsgebiet um ca. 1900 der französische Physiker HENRI BÉNARD (1874 - 1939). Die dabei beobachteten Vorgänge wurden 1916 von dem britischen Physiker LORD RAYLEIGH (1842 - 1919) anschließend theoretisch beschrieben.

Den Versuchsaufbau zur sogenannten *Rayleigh-Bénard-Konvektion* stelle man sich in etwa folgendermaßen vor:

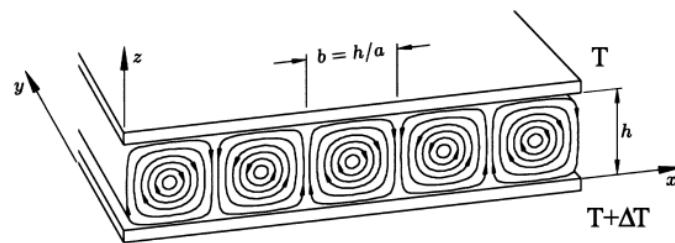


Abbildung 2.1: Versuchsaufbau Rayleigh-Bénard-Konvektion

Zwischen zwei Platten befindet sich, wie in Abb. 2.1<sup>1</sup> dargestellt, eine viskose, inkompressible Flüssigkeit. Durch Beheizung der unteren Platte kann dem System Wärmeenergie zugeführt werden.

Einen wichtigen Kontrollparameter dieses Experiments stellen die Intensität dieser Heizung und der entstehende Temperaturgradient dar. Als Reaktion des Systems können je nach Temperaturdifferenz  $\Delta T$  zwischen oberer und unterer Platte vier verschiedene Fälle eintreten:

1.  $\Delta T = 0$

Die Temperaturverteilung ist homogen, das System befindet sich im thermischen Gleichgewicht.

2.  $\Delta T < \Delta T_{cr}$

Der Verlauf von Dichte und Temperatur ist zwischen den beiden Platte linear, mittels Wärmeleitung wird dieser Inhomogenität entgegengewirkt. Das System ist stabil, es gibt keine makroskopischen Strömungen.

---

<sup>1</sup>Quelle: [19]

3.  $\Delta T \geq \Delta T_{cr}$

Die Temperatur- bzw. Dichtverteilung ist nicht mehr stabil. Bereits kleine Störungen wie Verschiebungen durch Molekularbewegungen führen zu makroskopischen Strömungen, das entstehende Konvektionsmuster ist jedoch zunächst regelmäßig und laminar. Der mehr oder minder zufällige Drehsinn der Konvektionsrollen hängt von diesen initiiierenden Fluktuationen ab, die den auslösenden Symmetriebruch bewirken.

Nicht nur bei Fluidexperimenten sondern auch in der realen Atmosphäre können außerdem immer wieder organisierte Konvektionszellen in bienenwabenähnlichen Formen beobachtet werden, siehe Abb. 2.2<sup>2</sup> und Abb. 2.3<sup>3</sup>.

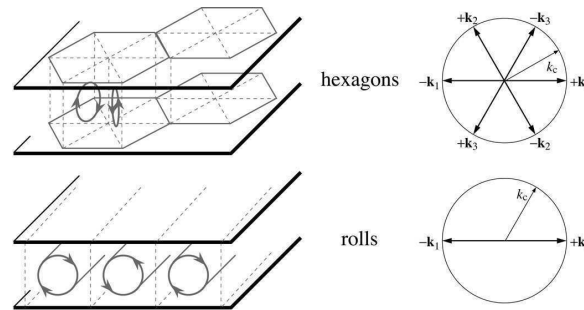


Abbildung 2.2: Hexagonale Konvektionsstrukturen und Konvektionsrollen

<sup>2</sup>Quelle: [24]

<sup>3</sup>Quelle: <http://eoimages.gsfc.nasa.gov/images/imagerecords/59000/59758/SouthAtlantic.A2002157.1055.1km.jpg>, 28.10.2012

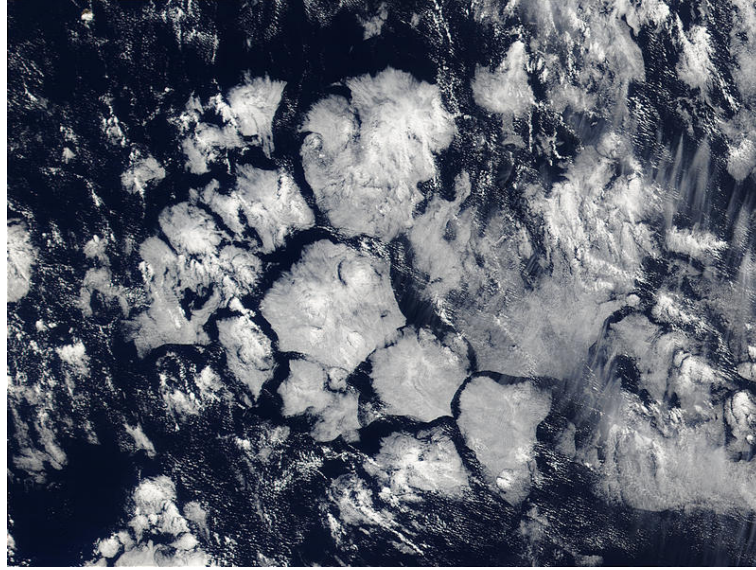


Abbildung 2.3: Satellitenbild Südatlantik

4.  $\Delta T \gg \Delta T_{cr}$

Mit steigender Temperaturdifferenz entwickelt sich ein im Verlauf zuerst periodisches, dann quasiperiodisches und schließlich turbulentes Strömungsbild.

### 2.1.3 LORD RAYLEIGH - Die Konvektionsgleichungen

Die theoretischen Grundlagen zur Herleitung der Rayleigh-Bénard Gleichungen sind direkt der Darstellung in [18] entnommen, sofern nicht anders angegeben.

Um die Bewegungsgleichung aus Kap. 2.1.1 zu modifizieren wird die Boussinesq-Approximation angewandt. Damit ist die Dichte überall als konstant ( $\rho = \rho_o$ ) anzusehen außer im Auftriebsterm, der für den Konvektionsmechanismus ausschlaggebend ist.

Nach Division durch  $\rho$  schreibt sich Gl. (2.1):

$$\frac{\partial \vec{v}}{\partial t} + \vec{v} \cdot \nabla \vec{v} = \frac{\rho}{\rho_0} \vec{g} - \frac{1}{\rho_0} \nabla p + \frac{\eta}{\rho_0} \nabla^2 \vec{v} \quad (2.4)$$

Die Dichte soll ausschließlich und linear von der Temperaturverteilung abhängen:

$$\rho(\vec{x}, t) = \rho_0 [1 - \alpha (T(\vec{x}, t) - T_0)] \quad (2.5)$$

$\alpha$  ist der Wärmeausdehnungskoeffizient.

Weiters wird die kinematische Viskosität  $\nu$  als Quotient der dynamischen Viskosität und der Dichte eingeführt:

$$\nu = \frac{\eta}{\rho_0} \quad (2.6)$$

Gl. (2.4) wird zu:

$$\frac{\partial \vec{v}}{\partial t} + \vec{v} \cdot \nabla \vec{v} = [1 - \alpha (T(\vec{x}, t) - T_0)] \vec{g} - \frac{1}{\rho_0} \nabla p + \nu \nabla^2 \vec{v} \quad (2.7)$$

Als weitere Vereinfachung wird angenommen, dass sich die Konvektionsrollen in  $x$ - $z$ -Richtung ausrichten, somit entfallen alle Abhängigkeiten von  $y$ , diverse Abhängigkeiten der Variablen werden nicht mehr in jeder Gleichung explizit angeschrieben:

$$\frac{du}{dt} = \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + w \frac{\partial u}{\partial z} = -\frac{1}{\rho_0} \frac{\partial p}{\partial x} + \nu \nabla^2 u \quad (2.8)$$

$$\frac{dw}{dt} = \frac{\partial w}{\partial t} + u \frac{\partial w}{\partial x} + w \frac{\partial w}{\partial z} = -\frac{1}{\rho_0} \frac{\partial p}{\partial z} + \nu \nabla^2 w + [1 - \alpha (T - T_0)] g \quad (2.9)$$

Einführung der Stromfunktion  $\psi$  und der Abweichung vom linearen Temperaturprofil  $\Theta$ :

$$(u, w) = \left( -\frac{\partial \Psi}{\partial z}, \frac{\partial \Psi}{\partial x} \right) \quad (2.10)$$

$$T(x, z, t) = T_0 + \bar{T} \left( 1 - \frac{z}{h} \right) + \Theta(x, z, t) \quad (2.11)$$

Durch  $\frac{\partial}{\partial x}$  (2.9) -  $\frac{\partial}{\partial z}$  (2.8) kann der Druck eliminiert werden:

$$\begin{aligned} \frac{\partial}{\partial t} \left( \frac{\partial w}{\partial x} - \frac{\partial u}{\partial z} \right) + \frac{\partial}{\partial x} \left( u \frac{\partial w}{\partial x} + w \frac{\partial w}{\partial z} \right) - \frac{\partial}{\partial z} \left( u \frac{\partial u}{\partial x} + w \frac{\partial u}{\partial z} \right) = \\ \nu \left( \frac{\partial}{\partial x} \nabla^2 w - \frac{\partial}{\partial z} \nabla^2 u \right) + \frac{\partial}{\partial x} [1 - \alpha (T - T_0)] g \end{aligned} \quad (2.12)$$

Einige Umformungen:

$$\left( \frac{\partial w}{\partial x} - \frac{\partial u}{\partial z} \right) = \left( \frac{\partial^2 \Psi}{\partial x^2} + \frac{\partial^2 \Psi}{\partial z^2} \right) = \nabla^2 \Psi \quad (2.13)$$

$$\frac{\partial}{\partial x} \left( u \frac{\partial w}{\partial x} + w \frac{\partial w}{\partial z} \right) - \frac{\partial}{\partial z} \left( u \frac{\partial u}{\partial x} + w \frac{\partial u}{\partial z} \right) =$$

$$u \left( \frac{\partial^2 w}{\partial x^2} - \frac{\partial^2 u}{\partial x \partial z} \right) + w \left( \frac{\partial^2 w}{\partial x \partial z} - \frac{\partial^2 u}{\partial x^2} \right) =$$

$$\frac{\partial \Psi}{\partial x} \frac{\partial}{\partial z} \nabla^2 \Psi - \frac{\partial \Psi}{\partial z} \frac{\partial}{\partial x} \nabla^2 \Psi = \frac{\partial (\Psi, \nabla^2 \Psi)}{\partial (x, z)} \quad (2.14)$$

wobei

$$\frac{\partial (a, b)}{\partial (x, z)} = \left( \frac{\partial a}{\partial x} \frac{\partial b}{\partial z} - \frac{\partial b}{\partial x} \frac{\partial a}{\partial z} \right) \quad (2.15)$$

weilers:

$$\begin{aligned} & \frac{\partial}{\partial x} \nabla^2 w - \frac{\partial}{\partial z} \nabla^2 u = \\ & \frac{\partial^3 w}{\partial x^3} + \frac{\partial^3 w}{\partial x \partial z^2} - \frac{\partial^3 u}{\partial x^2 \partial w} - \frac{\partial^3 u}{\partial z^3} = \\ & \nabla^2 \left( \frac{\partial w}{\partial x} - \frac{\partial u}{\partial z} \right) = \nabla^2 (\nabla^2 \Psi) = \nabla^4 \Psi \end{aligned} \quad (2.16)$$

und

$$\begin{aligned} & \frac{\partial}{\partial x} [1 - \alpha (T - T_0)] g = \\ & \frac{\partial}{\partial x} \left[ 1 - \alpha \left( \bar{T} \left( 1 - \frac{z}{h} \right) + \Theta \right) \right] g = \\ & g \alpha \frac{\partial \Theta}{\partial x} \end{aligned} \quad (2.17)$$

ergibt die Strömungsgleichung der *Rayleigh-Bénard-Konvektion*:

$\frac{\partial}{\partial t} \nabla^2 \Psi = - \frac{\partial (\Psi, \nabla^2 \Psi)}{\partial (x, z)} + \nu \nabla^4 \Psi + g \alpha \frac{\partial \Theta}{\partial x} \quad (2.18)$
---

Aus der Kontinuitätsgleichung (2.2) wird durch die Inkompressibilität:

$$\frac{\partial \rho}{\partial t} = 0 \quad \text{sowie} \quad \nabla \cdot \vec{v} = 0 \quad (2.19)$$

Auch die Energiegleichung wird angepasst:

Division durch  $(g c_p)$  und  $\rho = \rho_0$  führt unter Verwendung von

## 2 Das Lorenz(1963)-Modell in 3 Dimensionen

$$\kappa = \frac{\lambda}{\rho_0 c_p} \quad (2.20)$$

von (2.3) zu:

$$\frac{dT}{dt} = \frac{\partial T}{\partial t} + \vec{v} \cdot \nabla T = \kappa \Delta T \quad (2.21)$$

Wieder mit dem Temperaturfeld (2.11) kann diesmal die Temperaturadvektion umgeschrieben werden:

$$\begin{aligned} \vec{v} \cdot \nabla T &= \left( u \frac{\partial T}{\partial x} + w \frac{\partial T}{\partial z} \right) = \\ u \frac{\partial \Theta}{\partial x} + w \left( -\frac{\bar{T}}{h} + \frac{\partial \Theta}{\partial z} \right) &= \\ -\frac{\partial \Psi}{\partial z} \frac{\partial \Theta}{\partial x} + \frac{\partial \Psi}{\partial x} \left( -\frac{\bar{T}}{h} + \frac{\partial \Theta}{\partial z} \right) &= \\ \frac{\partial (\Psi, \Theta)}{\partial (x, z)} - \frac{\bar{T}}{h} \frac{\partial \Psi}{\partial x} \end{aligned} \quad (2.22)$$

Unter Berücksichtigung, dass im Temperaturfeld  $T$  nur die Fluktuationen  $\Theta$  von der Zeit abhängen, führt der Weg über

$$\frac{\partial T}{\partial t} = \frac{\partial \Theta}{\partial t} \quad (2.23)$$

und weiters

$$\Delta T = \Delta \Theta$$

zur zweiten wichtigen Rayleigh-Bénard-Gleichung:

$\frac{\partial \Theta}{\partial t} = -\frac{\partial (\Psi, \Theta)}{\partial (x, z)} + \frac{\bar{T}}{h} \frac{\partial \Psi}{\partial x} + \kappa \Delta \Theta \quad (2.24)$
--

Rayleigh fand heraus, dass sich für das Gleichungssystem, bestehend aus (2.18) und (2.24) unter geeigneter Parameterwahl Temperatur- und Strömungsfelder folgender Art einstellen können:

$$\Psi = \Psi_0 \sin \left( a\pi \frac{1}{h} x \right) \sin \left( \pi \frac{1}{h} z \right) \quad (2.25)$$

$$\Theta = \Theta_0 \cos \left( a\pi \frac{1}{h} x \right) \sin \left( \pi \frac{1}{h} z \right) \quad (2.26)$$

Dazu muss die nach ihm benannte Rayleigh-Zahl

$$Ra = g\alpha h^3 \Delta T \frac{1}{\nu\kappa} \quad (2.27)$$

einen bestimmten kritischen Wert übersteigen:

$$R_{cr} = \frac{\pi^4}{a^2} (1 + a^2)^3 \quad (2.28)$$

wobei  $a$  ein Geometriefaktor der Konvektionsrollen ist.

Der kleinste Wert  $R_{cr}$  bei dem dies eintreffen kann, kommt durch  $a^2 = \frac{1}{2}$  zustande und beträgt  $\frac{27\pi^4}{4}$ . [23]

## 2.1.4 BARRY SALTZMAN - Die Integration

Die theoretischen Grundlagen zur Integration der Rayleigh-Bénard-Gleichungen sind direkt der Darstellung in [32] entnommen.

Barry Saltzman beschäftigte sich nun mit der numerischen Lösung der Rayleigh-Bénard-Gleichungen, in seiner Arbeit “Finite Amplitude Free Convection as an Initial Value Problem”, erschienen im Juli 1962, beschreibt er seine Vorgangsweise.

Zunächst wählte er möglichst einfache Randbedingungen, wie freie Grenzflächen - *free / no stress boundaries* in  $z$ -Richtung bzw. gar keine Grenzen in  $x$ -Richtung. Obwohl diese Begriffe durchaus geläufig sind, sollen die physikalischen Konsequenzen hier trotzdem explizit erwähnt werden.

Horizontal: in  $x$ -Richtung wird die Ausdehnung als unendlich angesehen, es existieren also keine Grenzen und es werden bei der Integration der Gleichungen an diesen Rändern keine Bedingungen an die physikalischen Felder gestellt.

Vertikal: die Grenzen des Modells befinden sich bei  $z = 0$  (untere, beheizte Platte) und  $z = 1$  (obere Platte), es gilt:

$$\Psi|_{z=0} = \Psi|_{z=1} = 0 \quad (2.29)$$

$$\Theta|_{z=0} = \Theta|_{z=1} = 0 \quad (2.30)$$

daraus ergibt sich weiters:

$$\nabla^2 \Psi|_{z=0} = \nabla^2 \Psi|_{z=1} = 0 \quad (2.31)$$

$$\frac{\partial(\psi, \nabla^2 \psi)}{\partial(x, z)} \Big|_{z=0} = \frac{\partial(\psi, \nabla^2 \psi)}{\partial(x, z)} \Big|_{z=1} = 0 \quad (2.32)$$

$$\frac{\partial(\psi, \Theta)}{\partial(x, z)} \Big|_{z=0} = \frac{\partial(\psi, \Theta)}{\partial(x, z)} \Big|_{z=1} = 0 \quad (2.33)$$

Also sowohl Vorticity als auch deren Advektion sowie die Advektion der Temperaturstörung verschwinden an den vertikalen Rändern.

Er entwickelte die dynamischen Größen zur Beschreibung der Felder in doppelte Fourierreihen:

$$\Psi(x, y, z) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \Psi^*(m, n, t) \exp \left[ 2\pi h i \left( \frac{m}{l} x + \frac{n}{2h} z \right) \right] \quad (2.34)$$

$$\Theta(x, y, z) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \Theta^*(m, n, t) \exp \left[ 2\pi h i \left( \frac{m}{l} x + \frac{n}{2h} z \right) \right] \quad (2.35)$$

Dabei sind  $m$  und  $n$  die Wellenzahlen in  $x$ - und  $z$ -Richtung,  $l$  und  $h$  sind proportional zu den jeweiligen Wellenlängen.

Die komplexen Fourierkoeffizienten lassen sich dann berechnen:

$$\Psi^*(m, n, t) = \frac{1}{2lh} \int_0^L \int_{-h}^h \Psi(x, z, t) \exp \left[ -2\pi h i \left( \frac{m}{l} x + \frac{n}{2h} z \right) \right] dx dz \quad (2.36)$$

$$\Theta^*(m, n, t) = \frac{1}{2lh} \int_0^L \int_{-2}^h \Theta(x, z, t) \exp \left[ -2\pi h i \left( \frac{m}{l} x + \frac{n}{2h} z \right) \right] dx dz \quad (2.37)$$

Nachdem es sich bei diesen Reihen um Summationen unendlich vieler harmonischer Funktionen handelt, stellt sich an dieser Stelle die entscheidende Frage nach der Wahl der tatsächlich zu verwendenden Moden. Die abgeschnittenen Reihen müssen einerseits die nichtlineare Dynamik des physikalischen Modells enthalten, andererseits soll die Anzahl der Dimensionen im Wellenraum minimal gehalten werden.

Nach einer Reihe von mathematischen Operationen und Approximationen setzte Saltzman die entwickelten Größen in das Rayleigh-Bénard-Gleichungssystem ein. Zur numerischen Integration der verbleibenden 52 gewöhnlichen Differentialgleichungen verwendete er ein Vorwärtsschema in zwei Teilschritten (Heun-Schema) und entdeckte, dass bei der gewählten Parameterkonfiguration und mit den beschriebenen Randbedingungen alle Fourierkoeffizienten bis auf drei gegen Null streben.



### 2.1.5 EDWARD LORENZ - Die Lorenz-Gleichungen

Literatur: [23]

Edward Lorenz schloss direkt an die Resultate Saltzmans an, er verwendete die Kenntnis der ausschlaggebenden drei Komponenten um die Fourierreihen (2.34) und (2.35) abzuschneiden bevor er sie mit dem physikalischen Modell kombinierte. Nach Einführung der dimensionslosen dynamischen Variablen  $X = X(t)$ ,  $Y = Y(t)$  und  $Z = Z(t)$ , die genau diese drei Fourierkoeffizienten darstellen, konnte er das partielle Differentialgleichungssystem der Rayleigh-Bénard-Konvektion auf drei gewöhnliche Differentialgleichungen reduzieren.

Das Lorenz(1963)- Modell:

$$\frac{dX}{dt} = -\sigma X + \sigma Y \quad (2.38)$$

$$\frac{dY}{dt} = -XZ + rX - Y \quad (2.39)$$

$$\frac{dZ}{dt} = XY - bZ \quad (2.40)$$

### 2.1.6 Alternative Herleitung

Die Autorin ist um eine möglichst vollständig nachvollziehbare Herleitung bemüht, darum wird an dieser Stelle noch eine alternative Darstellung besprochen.

Nachdem die Blackbox der numerischen Integration und die Verwendung von sprachlichen Überleitungen wie “nach einer Reihe von mathematische Operationen” dem physikalischen Verständnis des Lesers nicht unbedingt dienlich sind, wird ein theoretischer Versuchsaufbau eingeführt, der die Realisierung der gewählten Randbedingungen darstellt.

Anstatt der Entwicklung von Fourierreihen werden direkt die physikalischen Größen linear approximiert.

Die theoretischen Grundlagen für diese alternative Herleitung sind direkt der Darstellung in [31] entnommen.

Man stelle sich den Versuchsaufbau vor, wie in Abb. 2.4 <sup>4</sup>gezeigt:

---

<sup>4</sup>Quelle: [31]

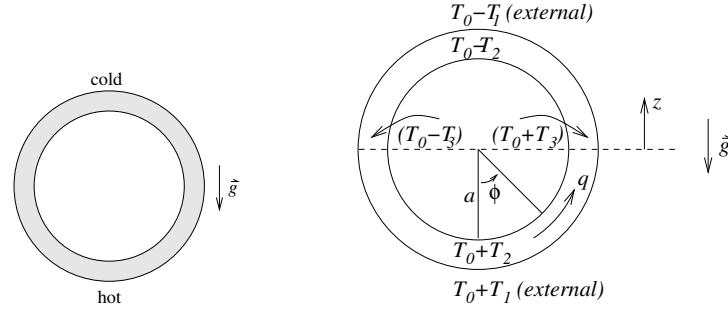


Abbildung 2.4: Versuchsaufbau zur Herleitung des 3-dimensionalen Lorenz(1963)-Modells

Das Fluid befindet sich nun gedanklich nicht mehr zwischen zwei Platten, sondern in einem Ring.

Der Vorteil dieser Konstruktion liegt unter anderem in der Geometrie, die durch ihre Nachempfindung der natürlichen Strömung in einer Konvektionsrolle eine eindimensionale Geschwindigkeitsbeschreibung zulässt. Sie entspricht somit einer Koordinatentransformation der Kartesischen zu Polarkoordinaten.

Der gesamte Radius des Ringes wird mit  $a$  bezeichnet, der innere Radius des Schlauchs soll viel geringer als dieser sein.  $\phi$  gibt die betrachtete Position an, im Mittelpunkt herrscht fiktiv die Temperatur  $T_0$ . Die Temperaturdifferenz beträgt zwischen höchstem und tiefstem Punkt  $2T_2$  und zwischen den Punkten ganz links und rechts außen  $2T_3$ .

Die äußere Temperatur ändert sich linear mit der Höhe:

$$T_E = T_0 - \frac{T_1 z}{a} = T_0 + T_1 \cos \phi \quad (2.41)$$

Obwohl sie in Realität viel komplexer ist, kann die Temperaturverteilung im Inneren vereinfacht dargestellt werden:

$$T - T_0 = T_2 \cos \phi + T_3 \sin \phi \quad (2.42)$$

wobei  $T_2 = T_2(t)$  und  $T_3 = T_3(t)$ .

Diese Näherungen des Temperaturprofils können natürlich nur deshalb so einfach verwendet werden, da das Endprodukt der Herleitung bereits bekannt ist, die Erhaltung der Modelldynamik könnte mit solch einer intuitiven Vorgehensweise ohne Vorwissen nicht gewährleistet werden.

Die dynamischen Variablen werden nun über die Querschnittflächen des Schlauchs gemittelt.

Geschwindigkeit:

$$q = q(\phi, t)$$

Temperatur (innerhalb des Schlauchs):

$$T = T(\phi, t)$$

Unter Verwendung von  $q$  als eindimensionale Geschwindigkeit und  $\phi$  als Raumkoordinate resultiert, vergleichbar mit (2.19), nach Anwendung der schon besprochenen Boussinesq-Approximation:

$$\frac{\partial \rho}{\partial t} = 0 \quad (2.43)$$

$$\frac{\partial q}{\partial \phi} = 0 \quad (2.44)$$

und somit:

$$q = q(t) \quad (2.45)$$

Die angepasste Impulsgleichung (2.4)

$$\frac{d\vec{v}}{dt} = -\frac{1}{\rho_0} \nabla p + \frac{\rho}{\rho_0} \vec{g} + \frac{\eta}{\rho_0} \nabla^2 \vec{v} \quad (2.46)$$

wird zu:

$$\frac{\partial q}{\partial t} = -\frac{1}{\rho a} \frac{\partial p}{\partial \phi} + g\alpha (T - T_0) \sin \phi - \Gamma q \quad (2.47)$$

Dazu wurden folgende Beziehungen verwendet:

- $\vec{v} \cdot \nabla \vec{v} = 0$  weil  $\frac{\partial q}{\partial \phi} = 0$
- $\nabla p = \frac{1}{a} \frac{\partial p}{\partial \phi}$  durch die Transformation in Polarkoordinaten
- Die Auftriebskraft  $g\alpha (T - T_0)$  wird mit dem Faktor  $\sin \phi$  erweitert um die Tangentialkomponente zu erhalten. Das Vorzeichen wird so bestimmt, dass das wärmere Fluid positiven Auftrieb erfährt.
- $\Gamma$  ist ein generalisierter Reibungskoeffizient, wichtig ist bei diesem Term die Proportionalität zur Geschwindigkeit  $q$ .

Einsetzen der Temperaturverteilung (2.42) führt zu:

$$\frac{\partial q}{\partial t} = -\frac{1}{\rho a} \frac{\partial p}{\partial \phi} + g\alpha (T_2 \cos \phi + T_3 \sin \phi) \sin \phi - \Gamma q \quad (2.48)$$

Um hier den Druckgradient-Term zu eliminieren wird einmal um den Ring von  $0 \rightarrow 2\pi$  integriert, dieser Schritt ist das Pendant zur Vorticitybildung durch die Rotation der Bewegungsgleichung im Kartesischen Koordinatensystem.

$$2\pi \frac{\partial q}{\partial t} = g\alpha \int_0^{2\pi} (T_2 \cos \phi \sin \phi + T_3 \sin^2 \phi) d\phi - 2\pi \Gamma q \quad (2.49)$$

$$\frac{dq}{dt} = -\Gamma q + \frac{g\alpha T_3}{2} \quad (2.50)$$

wobei  $\frac{\partial q}{\partial t}$  mit  $\frac{dq}{dt}$  ersetzt wird, da  $\frac{\partial q}{\partial \phi} = 0$ .

Es ist zu sehen, dass die horizontalen Temperaturdifferenz  $2T_3$  die Beschleunigung bewirkt.

Die angepasste Temperaturgleichung (2.21)

$$\frac{\partial T}{\partial t} + \vec{v} \cdot \nabla T = \kappa \Delta T \quad (2.51)$$

wird in Polarkoordinaten zu:

$$\frac{\partial T}{\partial t} + \frac{q}{a} \frac{\partial T}{\partial \phi} = \kappa (T_E - T) \quad (2.52)$$

Dazu wurden folgende Annahmen getroffen:

- Der Wärmetransport durch die Ringwand ergibt sich aus  $\kappa (T_E - T)$ .
- Die Wärmeleitung innerhalb des Fluids wird vernachlässigt.
- $\frac{q}{a} \frac{\partial T}{\partial \phi}$  ist hier ein Produkt gemittelter Größen, sollte genau genommen aber ein gemittelttes Produkt sein. Es wird also die Korrelation zwischen  $q$  und  $\frac{\partial T}{\partial \phi}$  nicht berücksichtigt.

Subtrahiert man (2.42) von (2.41), so ergibt sich:

$$T_E - T = (T_1 - T_2) \cos \phi - T_3 \sin \phi, \quad (2.53)$$

womit sich die Temperaturgleichung (2.52) umschreiben lässt zu:

$$\frac{dT_2}{dt} \cos \phi + \frac{dT_3}{dt} \sin \phi - \frac{q}{a} T_2 \sin \phi + \frac{q}{a} T_3 \cos \phi = \kappa (T_1 - T_2) \cos \phi - \kappa T_3 \sin \phi. \quad (2.54)$$

Dabei wurden die partiellen Ableitungen der Temperatur durch totale ersetzt, da  $T_2$  und  $T_3$  nur von der Zeit abhängig sind.

## 2 Das Lorenz(1963)-Modell in 3 Dimensionen

Da die Gleichung für alle  $\phi$  gültig sein muss, können die Sinus-Terme und die Cosinus-Terme separiert werden:

$$\sin \phi : \frac{dT_3}{dt} - \frac{qT_2}{a} = -\kappa T_3 \quad (2.55)$$

$$\cos \phi : \frac{dT_2}{dt} + \frac{qT_3}{a} = \kappa (T_1 - T_2) \quad (2.56)$$

(2.55) und (2.56) beschreiben zusammen mit Gl. (2.50) die Dynamik des Systems.

Um die Schreibweise zu verkürzen, wird  $T_4$  eingeführt:

$$T_4(t) = T_1 - T_2(t) \quad (2.57)$$

$T_4$  entspricht der Temperaturdifferenz zwischen innen und außen an der höchsten und tiefsten Position des Ringes, in Gl. (2.55) und (2.56) eingesetzt, führt dies zu:

$$\frac{dT_3}{dt} = -\kappa T_3 + \frac{qT_1}{a} - \frac{qT_4}{a} \quad (2.58)$$

$$\frac{dT_4}{dt} = -\kappa T_4 + \frac{qT_3}{a} \quad (2.59)$$

Nun werden für die Kurzschreibweise des Differentialgleichungssystems drei dimensionslose Variablen definiert:

$$X = \frac{q}{a\kappa} \quad (2.60)$$

$$Y = \frac{g\alpha T_3}{2a\Gamma\kappa} \quad (2.61)$$

$$Z = \frac{g\alpha T_4}{2a\Gamma\kappa} \quad (2.62)$$

Auch die Zeit wird entdimensioniert:

$$t' = \kappa t \quad (2.63)$$

Wird der Einfachheit halber  $t$  statt  $t'$  geschrieben und wieder die Rayleigh-Zahl aus Gl. (2.27)

$$Ra = \frac{g\alpha T_1}{2a\Gamma\kappa} \quad (2.64)$$

sowie die Prandtl-Zahl

$$\sigma = \frac{\Gamma}{\kappa} \quad (2.65)$$

verwendet, so führt dies zum Differentialgleichungssystem

$$\frac{dX}{dt} = -\sigma X + \sigma Y \quad (2.66)$$

$$\frac{dY}{dt} = -XZ + rX - Y \quad (2.67)$$

$$\frac{dZ}{dt} = XY - Z \quad (2.68)$$

Im bekannten Lorenz(1963)-Modell findet sich in der letzten Gleichung noch der Faktor  $b$ , er ist ähnlich einer horizontalen Wellenzahl im Konvektionsmuster zu deuten:

$$\frac{dZ}{dt} = XY - bZ \quad (2.69)$$

## 2.2 Eigenschaften

Es wird nun auf diverse Charakteristika des Systems eingegangen und die Bedeutung der eingeführten Variablen diskutiert.

Insgesamt lässt sich sagen, dass die Lorenz-Gleichungen die nichtlineare Dynamik einer einzelnen baroklinen Welle in Verbindung mit einer zeitlich variablen zonalen Strömung darstellen. [28]

### Autonomie

Das Lorenz(1963)-Modell ist ein autonomes Gleichungssystem, die Koeffizienten sind also nicht explizit von der Zeit abhängig. [23]

### Bedeutung diverser Größen

Literatur: [18, 23, 34, 35, 36]

- $X$  ist proportional zur Strömungsgeschwindigkeit in den Konvektionsrollen. Eine Änderung des Vorzeichens zeigt eine Umkehrung der Rotationsrichtung.

## 2 Das Lorenz(1963)-Modell in 3 Dimensionen

- $Y$  ist proportional zur Temperaturdifferenz zwischen aufsteigenden und absinkenden Volumselementen.
- $Z$  ist proportional zur Differenz vom vertikalen linearen Temperaturprofil.

Bei reiner Wärmeleitung stellt sich ein Temperaturfeld horizontal homogen und vertikal linear ein,  $Y$  und  $Z$  beschreiben die Abweichung zu diesem.

Ist das Produkt von  $X$  und  $Y$  positiv, so sind die aufsteigenden Fluidmassen wärmer als die absinkenden.

- $b$  beschreibt die Geometrie der Konvektionszellen. Für das Einsetzen der Konvektion beim kleinstmöglichen vertikalen Temperaturgradienten wird oft der Wert  $b = \frac{3}{8}$  gewählt.
- $\sigma$  ist die dimensionslose *Prandtl-Zahl* und beschreibt die Trägheit eines Systems. Sie setzt die Transporteigenschaften eines Fluids bezüglich Impuls ( $\nu$  - kinematische Viskosität in  $\left[\frac{m^2}{s}\right]$ ) und Wärme ( $\kappa$  - Temperaturleitfähigkeit in  $\left[\frac{m^2}{s}\right]$ ) in Verhältnis.

$$\sigma = \frac{\nu}{\kappa} \sim \frac{\text{durch innere Reibung erzeugte Wärme}}{\text{durch Strömung transportierte Wärme}}$$

Oft wird der Wert  $\sigma = 10$  (entspricht kaltem Wasser) verwendet.

- $r$  ist die relative Rayleigh-Zahl

$$r = \frac{R_a}{R_{cr}} \quad (2.70)$$

wobei

$$Ra = g\alpha h^3 \Delta T \frac{1}{\nu\kappa}, \quad (2.71)$$

$$R_{cr} = \frac{\pi^4}{a^2} (1 + a^2)^3 \quad (2.72)$$

und  $a$  ebenfalls der Geometriebeschreibung der Konvektionsrollen dient.

Die Rayleigh-Zahl ist anschaulich beschrieben das Verhältnis zwischen auftreibender und abbremsender Kraft. Realisiert wird sie durch die Beheizung des Systems. Realistische Ergebnisse liefert das Lorenz(1963)-Modell für den Bereich  $R_a \simeq R_{cr}$ , also  $r \simeq 1$ , hier befindet sich die Grenze, ab der sich konvektive Strömung entwickelt.

Das Verhalten des Modells kann demnach durch die Wahl dieses Parameters gesteuert werden, eine detaillierte Dokumentation über den zeitlichen Verlauf der einzelnen Komponenten für verschiedene Rayleigh-Zahlen kann in [34] nachgelesen werden.

## Phasenraum & Attraktor

Für die Betrachtung des Systemverhaltens wird das Konzept des Phasenraums verwendet, das Lorenz in seiner Arbeit [23] wie folgt beschreibt:

Man stelle sich ein System vor, dessen Zustand durch  $M$  Variablen  $X_1, \dots, X_M$  gegeben ist. Die zeitliche Entwicklung wird durch das Gleichungssystem

$$\frac{dX_i}{dt} = F_i(X_1, \dots, X_M), \quad i = 1, \dots, M \quad (2.73)$$

vorgegeben, wobei die Zeit  $t$  die einzige unabhängige Variable ist und die Funktionen  $F_i$  stetige, erste partielle Ableitungen besitzen. Solch ein System kann dann in einem  $M$ -dimensionalen Euklidischen Raum, dem Phasenraum, dessen Koordinaten  $X_1, \dots, X_M$  sind, untersucht werden.

Jeder Punkt im Phasenraum entspricht einem möglichen Zustand des Systems. Die Entwicklung dieses Zustands entsprechend (2.38), (2.39) und (2.40) wird von einem Partikel repräsentiert, das sich entlang einer Trajektorie durch den Phasenraum bewegt.

Unter Verwendung entsprechend gewählter Systemparameter (wie oben beschrieben) lässt diese Trajektorie nach längerer Integration der Lorenz-Gleichungen den sogenannten *Lorenz-Attraktor* erkennen. In Abb. 2.5<sup>5</sup> und Abb. 2.6<sup>6</sup> ist der Attraktor im 3-dimensionalen Phasenraum bzw. in mehreren 2-dimensionalen Schnittebenen dargestellt. Der Attraktor ist ein Unterraum des Phasenraums und beinhaltet alle Zustände, die tatsächlich durch die konkreten Entwicklungsgleichungen realisiert werden können.

---

<sup>5</sup>Quelle: Code auf Seite 160

<sup>6</sup>Quelle: Code auf Seite 160



## 2 Das Lorenz(1963)-Modell in 3 Dimensionen

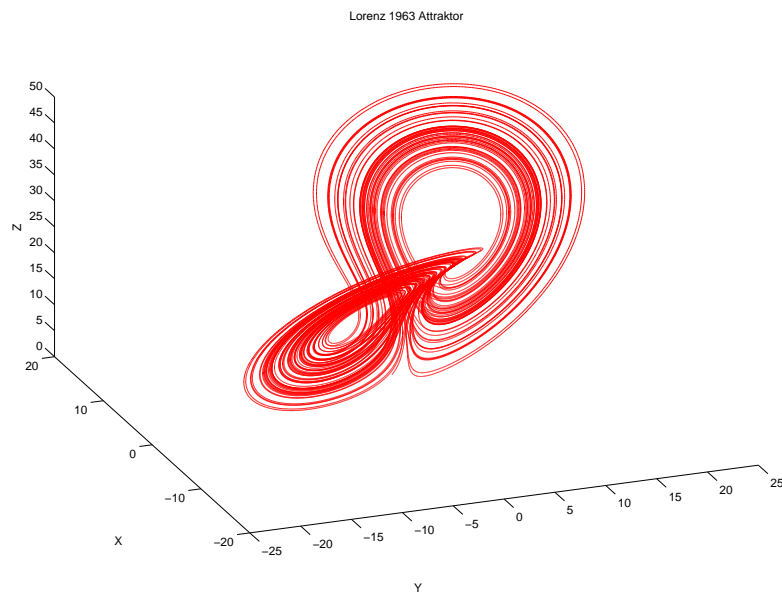


Abbildung 2.5: Lorenz(1963)-Attraktor im 3-dimensionalen Phasenraum

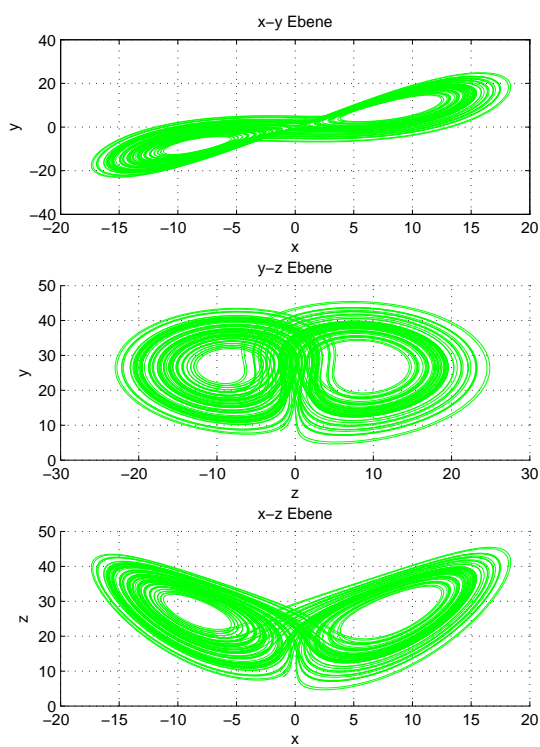


Abbildung 2.6: Lorenz(1963)-Attraktor in 2-dimensionalen Schnittebenen

## Physikalische Interpretation des Attraktors

Literatur: [18, 23, 31, 34]

Der Nullpunkt des Phasenraumes kennzeichnet den Systemzustand, bei dem sich keine Konvektion einstellt, die Temperaturverteilung ist entweder homogen oder die Gradienten so schwach, dass reine Wärmeleitung stattfindet.

Jeweils in den Spiralmittelpunkten befinden sich die Fixpunkte der stationären Lösungen, hier ist die Konvektion regelmäßig und die Strömungsgeschwindigkeit bleibt konstant. Die Bewegung rund um diese Fixpunkte und der aperiodische Wechsel zwischen den beiden Spiralen zeigen den Zustand turbulenter Konvektion. Auf welcher der beiden Spiralen des Attraktors sich der aktuelle Zustand lokalisiert, hängt mit dem Drehsinn der Konvektionsrollen zusammen.

Es zeigt sich hier die *Symmetrie* des physikalischen Modells, in dem die beiden möglichen Rotationsrichtungen als gleichwertig anzusehen sind. Wird der Zustandsvektor  $(X, Y, Z)$  durch  $(-X, -Y, Z)$  ersetzt, so ändern sich im Gleichungssystem ebenfalls nur die Vorzeichen der ersten beiden Komponenten, die  $Z$ -Komponente bleibt gleich. Somit ist jede Trajektorie symmetrisch oder besitzt ein Gegenstück, das durch Spiegelung an der  $Z$ -Achse erzeugt werden kann.

Numerische Berechnungen zeigen, dass Trajektorien mit Startpunkt außerhalb des Attraktors sehr schnell zu diesem finden.

Die Bahnen der Trajektorien kreuzen einander zu keinem Zeitpunkt und passieren bei turbulenter Konvektion natürlich auch weder den Nullpunkt, noch einen der anderen beiden Fixpunkte.

Um die Gleichgewichtszustände zu ermitteln müssen die zeitlichen Entwicklungen jeder Komponente Null gesetzt werden:

$$\frac{dX}{dt} = \frac{dY}{dt} = \frac{dZ}{dt} = 0 \quad (2.74)$$

Das Gleichungssystem wird zu:

$$0 = -\sigma X + \sigma Y \quad (2.75)$$

$$0 = -XZ + rX - Y \quad (2.76)$$

$$0 = XY - bZ \quad (2.77)$$

Die triviale Lösung  $X = Y = Z = 0$  beschreibt den konvektionsfreien Zustand.

Weitere Lösungen sind:

$$X_{2,3} = \pm \sqrt{(r-1)b}$$

$$Y_{2,3} = \pm \sqrt{(r-1)b}$$

$$Z_{2,3} = r - 1,$$

die jedoch nur für  $r \geq 1$  Fixpunkte darstellen, da Werte für  $r < 1$  durch die negative Diskriminante der Wurzel nicht verschwindende Imaginärteile bewirken würden.

Die Punkte  $(X, Y, Z)_{2,3}$  sind die Zustände der regelmäßigen, konstanten Konvektionströmung.

Um Aussagen über die Stabilität der drei Gleichgewichtszustände machen zu können, müssen die Eigenwerte der Gleichungsmatrix berechnet werden. Dies wird zum Beispiel in [31] ausführlich durchgeführt und bringt für  $b=1$  folgende Resultate:

$$X = Y = Z = 0 \quad \begin{cases} \text{stabil für} & 0 < r < 1 \\ \text{instabil für} & r > 1 \end{cases} \quad (2.78)$$

$$X = Y = \pm \sqrt{(r-1)b}, \quad Z = r - 1 \quad \begin{cases} \text{stabil für} & 1 < r < r_c \\ \text{instabil für} & r > r_c \end{cases} \quad (2.79)$$

## Deterministisches Chaos

Die Dimension des Lorenz-Attraktors ist einerseits kleiner als die des 3-dimensionalen Phasenraums, er ist aber andererseits auch keine 2-dimensionale Fläche. Bei genauerer Analyse kann gezeigt werden, dass die Kapazitätsdimension des dissipativen Systems tatsächlich 2.06 ist. [19]

Im Unterschied zu geometrisch einfach beschreibbaren Attraktoren wie z.B. Fixpunkten, Grenzzyklen oder Tori, ist er damit ein sogenannter *seltsamer Attraktor*.

Diese werden in [38] sehr treffend beschrieben und fassen somit einige wichtige Eigenschaften des Lorenz(1963)-Modells zusammen:

„Man spricht vom seltsamen Attraktor wenn das System im Endzustand ein nicht periodisches Verhalten zeigt, sich nicht in einer geschlossenen geometrischen Form beschreiben lässt und keine ganzzahlige Dimension besitzt. Die Rede ist von einem Fraktal. Besonders auffällig ist das chaotische Verhalten, welches die Folge ist, dass jede kleine Änderung des Anfangszustandes den gesamten Verlauf grundlegend verändert. Seltsame Attraktoren liefern die mathematische Grundlage zur Beschreibung chaotischer Vorgänge wie etwa turbulenten Strömungen.“

Um den scheinbaren Widerspruch zwischen Chaos und Determinismus zu klären, werden die Definitionen der Begriffe Chaos, Determinismus und Deterministisches Chaos aus einem Skript der Technischen Universität Dortmund [41] angegeben:

- „*Chaos*:  
ursprünglich altgriechisch: der ungeordnete Urstoff vor der Weltschöpfung  
hier: schwer vorhersehbarer Zustand
- *Determinismus*:  
Ein Algorithmus [sic] heißt deterministisch, wenn es zu jeder (Programm-)Situation höchstens eine nachfolgende Situation geben kann, wenn also zu jeden Zeitpunkt der Folgeschritt eindeutig bestimmt ist.
- *Deterministisches Chaos*:  
Verhalten, bei dem einfache deterministische Gesetze zu irregulären Bewegungen führen, wird als Deterministisches Chaos bezeichnet.”

### *Lyapunov-Exponent:*

Um die chaotische Eigenschaft eines Differentialgleichungssystems mathematisch zu konkretisieren, wird das Konzept des Lyapunov-Exponenten verwendet.

Es wird berechnet ob Störungen, also leicht veränderte Anfangszustände im weiteren Verlauf konvergieren, divergieren oder äquidistant verlaufen, hierzu ist eine Betrachtung der Systementwicklung über einen längeren Zeitraum notwendig. Die Berechnung des Lyapunov-Exponenten  $\lambda$  erfolgt durch:

$$\lambda = \lim_{t \rightarrow \infty} \lim_{|\Delta x_0| \rightarrow 0} \frac{1}{t} \ln \left[ \frac{|\Delta x(X_{0,t})|}{|\Delta x_0|} \right] \quad (2.80)$$

Dabei ist  $\Delta x$  die Störung,  $x_0$  die Störung zum Zeitpunkt  $t = 0$  und  $X_0$  der Systemzustand zum Zeitpunkt  $t = 0$ .

$\Delta x$  ist abhängig vom Grundzustand zum Startzeitpunkt, da die Entwicklung der Störung linear aus dem aktuellen Systemzustand berechnet wird, welchem selbst das nichtlineare Gleichungssystem zugrunde liegt.

- $\lambda < 0$  :  
Das System nähert sich einem stabilen Fixpunkt im Phasenraum oder einem periodisch schwankenden Zustand, es wird als *asymptotisch stabil* bezeichnet.
- $\lambda = 0$  :  
Das System ist stationär, es wird als *lyapunov-stabil* bezeichnet.
- $\lambda > 0$  :  
Das System ist *instabil* und chaotisch, leicht unterschiedliche Anfangszustände divergieren.

Die Lyapunov Analyse wird komponentenweise durchgeführt und das Modell als chaotisch bezeichnet, sobald einer der Lyapunov-Exponenten positiv ist. [35]

## 2 Das Lorenz(1963)-Modell in 3 Dimensionen

Das Lorenz(1963)-Modell liefert unter Verwendung der oben besprochenen Parameter (Code auf Seite 160) nach ausreichend langer Integration folgende Werte:

Lyapunov Spektrum:

2.302585

2.347695

1.952947

Das Modell kann damit eindeutig als chaotisch identifiziert werden.

Die angesprochene Sensibilität auf leicht unterschiedliche Anfangszustände wird in Abb. 2.7<sup>7</sup> dargestellt :

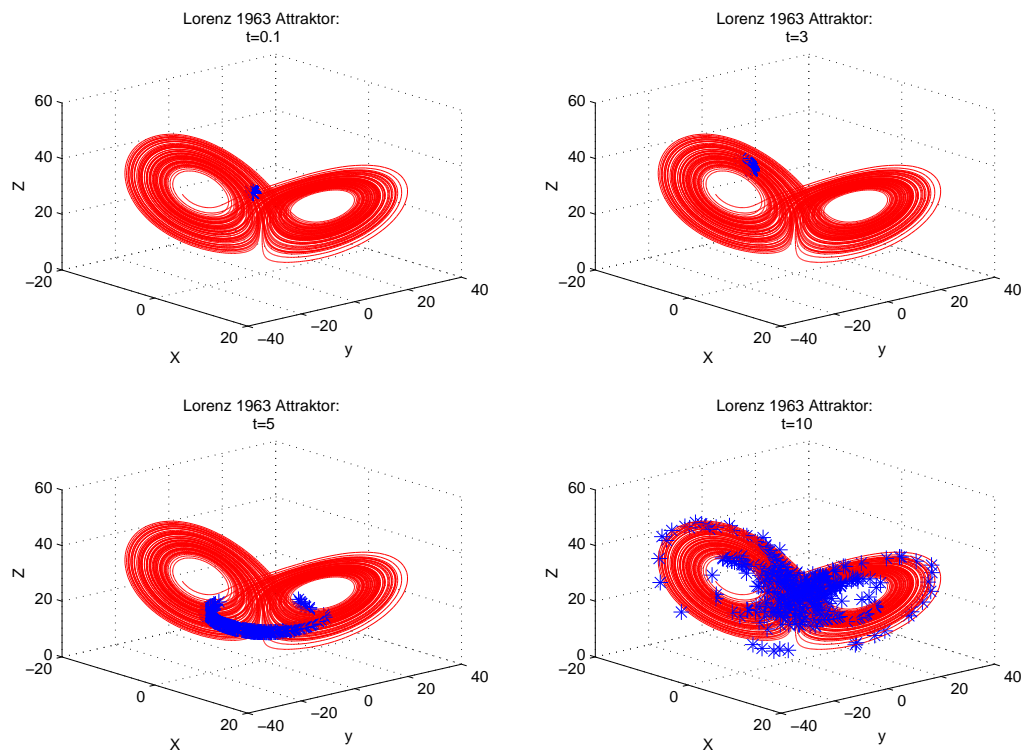


Abbildung 2.7: Lorenz(1963)-Attraktor: Störungsdivergenz

Zu sehen ist die Divergenz von Trajektorien, die im Anfangszustand sehr nahe beieinander liegen. Dazu wurden 500 Störungen eines Zustandes generiert und deren Entwicklung zu den Zeitpunkten  $t = 0.1, 3, 5$  und  $10$  dargestellt. Zu Beginn der Integration ist eine

---

<sup>7</sup>Quelle: Code auf Seite 164

Punktwolke von Zuständen im Phasenraum zu erkennen, während der zeitlichen Entwicklung verteilen sich die Endzustände früher oder später über den gesamten Attraktor. Beliebige kleine Störungen im Anfangszustand können so nach einer gewissen Zeit  $t$  überall am Attraktor liegende Endzustände erreichen. Diese Sensitivität bezüglich der Anfangsbedingungen ist typisch für ein chaotisches System. [18]

Wird das Wachstum möglicher Störungen (singuläre Vektoren der Entwicklungsmatrix) anhand der Singulärwerte untersucht, so ist festzustellen, dass diese nicht überall auf dem Attraktor (Abb. 2.8<sup>8</sup>) gleich sind. Es gibt Zustände, sprich Regionen am Attraktor, die auf Störungen sensibler reagieren als andere. [11]

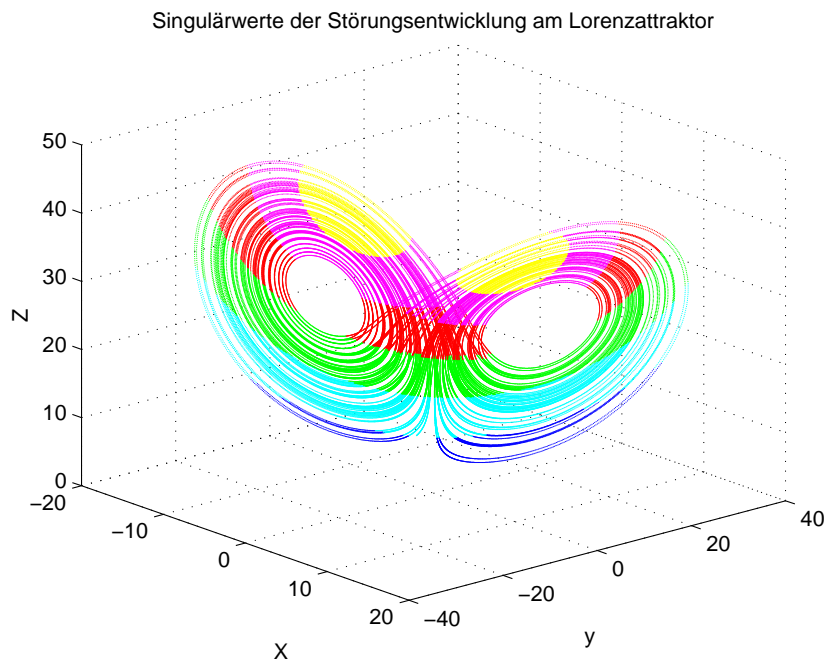


Abbildung 2.8: Lorenz(1963)-Attraktor: Sensitivität bezüglich Störungen im Anfangszustand

### Nichtlinearität

Wie schon eingangs erwähnt, ist es bei der Vereinfachung der Rayleigh-Bénard-Gleichung zur Einführung eines finite-mode models wichtig, den nichtlinearen Charakter des Modells zu bewahren. Nur durch die Wechselwirkung der einzelnen Komponenten ist es möglich trotz konstantem Forcing genau die aperiodischen Lösungen zu erhalten, die die Eigenschaften der zugrunde liegenden physikalischen Gleichungen darstellen. [23]

---

<sup>8</sup>Quelle: Code auf Seite 168

## Dissipation

Literatur: [18, 19, 23, 31, 34]

Das Lorenz(1963)-Modell hat einem konservativen Kern:

$$\begin{aligned}\frac{dX}{dt} &= \sigma Y \\ \frac{dY}{dt} &= -XZ + rX \\ \frac{dZ}{dt} &= XY\end{aligned}$$

In seiner vollständigen Form

$$\begin{aligned}\frac{dX}{dt} &= -\sigma X + \sigma Y \\ \frac{dY}{dt} &= -XZ + rX - Y \\ \frac{dZ}{dt} &= XY - Z\end{aligned}$$

ist es jedoch dissipativ, nur aufgrund der ständigen Energiezufuhr kommt es nicht zum Erliegen.

Es könnte sich die berechtigte Frage stellen, wo diese ständige Energiezufuhr in den Evolutionsgleichungen zu finden ist, da ein expliziter Forcing-Term nicht ausgemacht werden kann.

Für die Aufklärung dieses Mysteriums muss die Herleitung der Rayleigh-Bénard-Gleichungen genauer betrachtet werden: Zwar existieren Gleichungen für die Temperaturentwicklung und -verteilung für das Innere des Versuchsaufbaus, allerdings nicht für die äußere Umgebung. Da aber vom Fluid die Wärme der unteren Platte eindeutig abgeführt wird und auch der Wärmeverlust durch die Wände des Aufbaus nicht vernachlässigt wird, muss diese permanent geheizt werden. Mangels fehlender Entwicklungsgleichungen hat sie sozusagen gar nicht die Möglichkeit, auszukühlen.

Beide Attraktoren sind in Abb. 2.9<sup>9</sup> zu sehen.

---

<sup>9</sup>Quelle: Code auf Seite 171

## 2 Das Lorenz(1963)-Modell in 3 Dimensionen

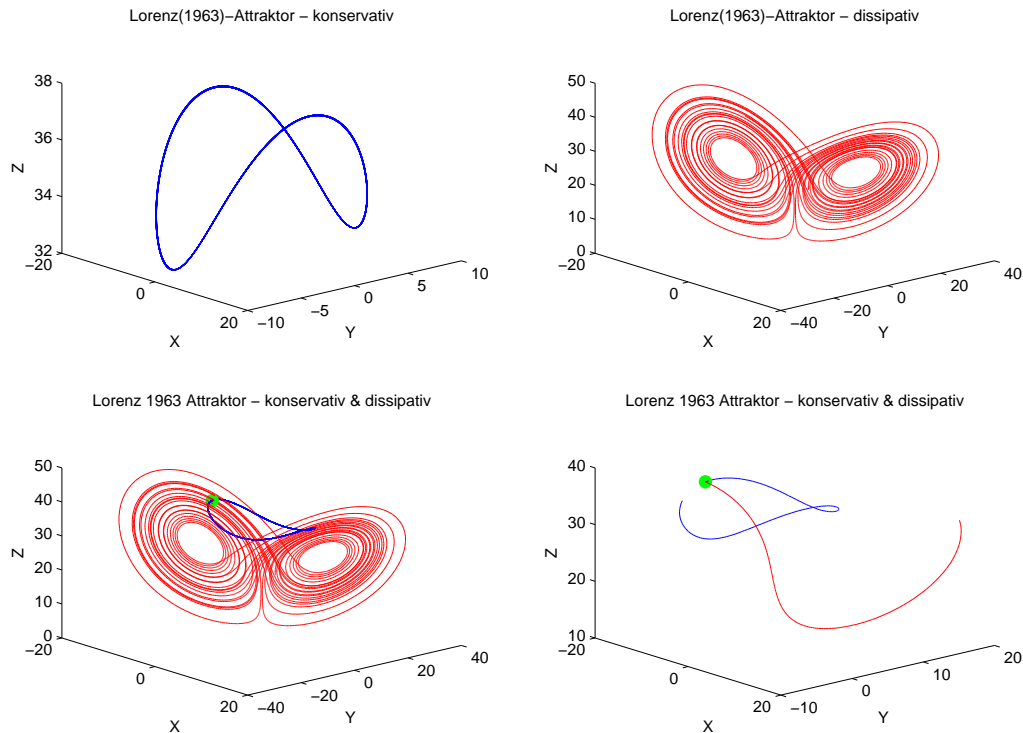


Abbildung 2.9: Lorenz(1963)-Attraktor: konservativ und dissipativ

Der Plot rechts unten zeigt, wie schnell die Trajektorien der beiden Attraktoren divergieren.

### Aufenthaltswahrscheinlichkeit des Systemzustands

Wie bereits beschrieben, kann jeder Zustand und seine Entwicklung durch ein Partikel und seine Trajektorie im Phasenraum dargestellt werden. Ist der Zustand des Systems nicht sicher anzugeben, so kann er anstatt durch einen einzelnen Punkt durch seine Aufenthaltswahrscheinlichkeit beschrieben werden. Diese Aufenthaltswahrscheinlichkeit nimmt ein Volumen im Phasenraum ein und ist begrenzt durch eine unter Umständen mehrdimensionale Isoplethe.

Zu Beginn befinden sich alle Zustände/Partikel, die mit einer gewissen Wahrscheinlichkeit "wahr" sind, innerhalb des Volumens. Wird das Differentialgleichungssystem integriert, so folgt jeder Zustand seiner Entwicklung/ jedes Partikel seiner Bahn. Am Ende sind noch immer alle Zustände/Partikel in diesem Volumen, ihre Partikelbahnen können sich weder kreuzen noch das Volumen verlassen. Dieses kann sich allerdings im Phasenraum bewegen und auch verformen bzw. seine Größe ändern.



Nimmt das Volumen zu, so sinkt die Wahrscheinlichkeitsdichte des Zustands und die Unsicherheit der Vorhersage wächst.

Zur Untersuchung dieser Volumsänderung wird die *Phasenraumdivergenz* des vollständigen Systems analysiert, direkt übernommen aus [31]:

$$\frac{d\dot{X}}{dt} + \frac{d\dot{Y}}{dt} + \frac{d\dot{Z}}{dt} = -\sigma - 1 - b \quad (2.81)$$

wobei  $\dot{X}, \dot{Y}$  und  $\dot{Z}$  jeweils die zeitlichen Ableitungen  $\frac{dX}{dt}, \frac{dY}{dt}$  und  $\frac{dZ}{dt}$  darstellen .

Die Divergenz der Gleichungsmatrix ist eindeutig negativ da  $\sigma$  und  $b$  beide  $> 0$  sind.

So lässt sich die zeitliche Änderung der Volumsgröße in seiner differentiellen Form

$$\frac{dV}{dt} = -(\sigma + 1 + b) V \quad (2.82)$$

schreiben und führt zu einer Lösung der Art:

$$V(t) = V(0) \exp [-(\sigma + 1 + b) t] \quad (2.83)$$

Dieses Ergebnis zeigt eine Volumskontraktion, womit der dissipative Charakter des Modells unabhängig des aktuellen Zustandes  $(X, Y, Z)$  eindeutig bewiesen werden kann.

Doch wie lässt sich diese globale Eigenschaft der Konvergenz mit der zuvor gezeigten Divergenz der gestörten Partikelbahnen vereinbaren, die ja für das chaotische System maßgeblich ist?

Werden die 3 Komponenten  $-\sigma, -1$  und  $-b$  als Koordinaten aufgefasst, so kann das System rotiert werden, sodass eine Komponente positiv wird, damit wird das Vorzeichen der Divergenz formal verändert oder eine Nulldivergenz erreicht.

Wird zum Beispiel ein synoptisches Dilatationsfeld mit einer Divergenz von Null betrachtet, so können sich die Punkte auf der Dilatationsachse trotzdem voneinander entfernen, auch für Divergenzen kleiner Null funktioniert das noch.

# 3 Alternative Formulierungen der klassischen Mechanik

Soweit nicht anders angegeben stammen die theoretischen Grundlagen für Kap. 3 aus dem Vorlesungsskript [3] “Ausgewählte Themen der dynamischen Meteorologie”, 2011 und dem auch dafür verwendeten Buch [29] “Grundkurs Theoretische Physik 2”, 2010. Diverse Definitionen, mathematische Sätze und manche Erklärungen wurden direkt übernommen.

Nachdem die Herleitung des Lorenz(1963)-Modells im letzten Kapitel eher numerisch und intuitiv begründet war, wird nun ein theoretischer Standpunkt gewählt.

Mit dem Kalkül der Hamiltonschen Mechanik und deren Erweiterung zur Nambu-Mechanik soll das System mathematisch rigoroser beurteilt und auf diverse Unstimmigkeiten aufmerksam gemacht werden.

Der in Kap. 2.1 verwendete Ansatz zur Herleitung des reduzierten Modells wird als Galerkin-Verfahren bezeichnet:

Die dynamischen Variablen des partiellen Differentialgleichungssystems, der Rayleigh-Bénard-Konvektionsgleichungen, wurden zunächst in Fourierreihen entwickelt. Diese unendlichen Reihen wurden dann mit Hilfe von Vorkenntnissen numerischer Ergebnisse abgeschnitten und in die Originalgleichungen zurück eingesetzt. Somit entstand ein gewöhnliches Differentialgleichungssystem, das die Eigenschaften des vollständigen physikalischen Problems widerspiegeln und für weitere Fragestellungen untersucht werden sollte.

Der sensible Punkt, die Schwierigkeit bei der Entwicklung eines finite-mode models liegt naturgemäß in der Wahl der notwendigen Moden, also in der Abschneidung dieser Fourierreihen, für die es, soweit im Moment bekannt, kein allgemein gültiges Kriterium gibt. Es fehlt demnach jegliche Garantie für die Erhaltung der Systemeigenschaften unter der vorgenommenen Reduktion.

Um den strukturellen Charakter, also zum Beispiel Nichtlinearitäten höherer Ordnung, Erhaltung physikalisch relevanter Größen wie Energie oder Vorticity und die Hamiltonsche Form der Ausgangsgleichungen zu bewahren, gibt es Ansätze, deren Verwendung

jedoch einer kurzen Einführung in alternative Formalismen der klassischen Mechanik bedarf. [7]

Für mathematisch vollständige Formulierungen und Beweise diverser Sätze wird der Leser an dieser Stelle dazu angehalten, in Werken der klassischen Mechanik nachzuschlagen, da hier nur das Gerüst der wichtigsten Meilensteine zum allgemeinen Verständnis aufgestellt werden soll.

Dafür empfohlen werden unter anderem:

- Nolting W.: Grundkurs Theoretische Physik 2, Springer Verlag, 2004
- Scheck, F.: Theoretische Physik 1: Klassische Mechanik, Springer Verlag, 2007
- Arnol'd, V. I.: Mathematical methods of classical mechanics, Springer Verlag, 1989
- Olver P. J.: Application of Lie Groups to Differential Equations, Springer Verlag, 2000

## 3.1 Mathematische Modelle der klassischen Mechanik

### 3.1.1 Mechanik nach Newton

Der Newtonsche Formalismus ist die wohl bekannteste Beschreibungsweise der diskreten klassischen Punktmechanik. Jede Beschleunigung entsteht durch die Summe aller angreifenden Kräfte:

$$m \frac{d^2 x}{dt^2} = \sum_i F_i \quad (3.1)$$

### 3.1.2 Lagrange-Mechanik

Der Einführung in die Lagrange-Mechanik seien einige mathematische Grundlagen vorangestellt.

#### Funktionale

Definition: *Funktional*

Ein Funktional ist eine Abbildung von einem Vektorraum von Funktionen in den Körper der reellen Zahlen.

Ein Funktional ist demnach eine Funktion, die als Argument ebenfalls eine Funktion hat. Wird mit einer Funktion beispielsweise jedem Wert  $x$  ein reeller Wert  $F(x)$  zugeordnet, so wird mit einem Funktional jeder Funktion  $F(x)$  ein reeller Wert  $\mathcal{F}[F(x)]$  zugeordnet.

Die in diesem Zusammenhang verwendete Klasse von Funktionalen sind die *Integrale*:

$$\mathcal{F}[x] = \int_{t_0}^{t_1} F(t, x(t), \dot{x}(t)) dt \quad (3.2)$$

wobei  $t$  die Zeit,  $x(t)$  eine Bahnkurve,  $\dot{x}(t)$  die Bahngeschwindigkeit und  $F(t, x(t), \dot{x}(t))$  eine Funktion dieser Größen darstellt. Für jede Bahnkurve wird durch die Integration eine dazugehörige reelle Zahl ermittelt.

Ein anschauliches Beispiel hierfür wäre die Verbindung zweier Punkte, deren Weglänge je nach gewählter Bahn variiert.

#### Auffinden von Extremalen eines Funktionals

In der *Variationsrechnung* ist häufig das Berechnen von Extremalen eines Funktionals gefragt, vergleichbar mit gewöhnlichen Extremwertberechnungen stellt sich hier die Frage nach der Funktion  $F(t, x(t), \dot{x}(t))$  für die das Funktional  $\mathcal{F}[F(x)]$  minimal oder maximal wird.

$$\int_{t_0}^{t_1} F(t, x(t), \dot{x}(t)) dt = \text{Extr.} \quad (3.3)$$

Im zuvor erwähnten anschaulichen Beispiel kann auf diese Weise der kürzeste Weg ermittelt werden.

Durch Rückführung dieses Problems auf eine gewöhnliche Extremwertberechnung mit Hilfe der Konstruktion einer um  $\varepsilon$  von der Bahnkurve abweichenden Funktionenschar, die an den zeitlichen Randpunkten fixiert wird, kann gezeigt werden, dass die Funktion  $F(x)$  ein Extremum liefert, wenn gilt:

$$\frac{\partial F}{\partial x} - \frac{d}{dt} \frac{\partial F}{\partial \dot{x}} = 0 \quad (3.4)$$

Diese Gleichung wird als *Euler-Lagrange-Gleichung* bezeichnet.

## Hamiltonsches Prinzip

Satz: *Hamiltonsches Prinzip*

Die Bewegung eines mechanischen Systems (3.4) stimmt mit den Extremalen des Funktionals

$$S[\vec{q}(t)] = \int_{t_0}^{t_1} L(\vec{q}, \dot{\vec{q}}, t) dt \quad (3.5)$$

überein, wenn

$$L(\vec{q}, \dot{\vec{q}}, t) = T - U \quad (3.6)$$

die Differenz zwischen *kinetischer* und *potentieller Energie* des Systems ist.

Definition: *Lagrange-Funktion*  $L(\vec{q}, \dot{\vec{q}}, t)$ :

Hat man ein System von  $n$  Punktteilchen im dreidimensionalen Raum vorliegen, so ist  $\vec{q} = (q_1, \dots, q_{3n})$ . Im Allgemeinen müssen die  $q_i$  jedoch keine Lagekoordinaten sein, z.B. könnte ein System auch durch die Angabe von Winkeln bestimmt sein. Aus diesem Grund bezeichnet man die  $q_i$  als *generalisierte Koordinaten*. Die zugehörige Zeitableitungen  $\dot{q}_i$  sind die *generalisierten Geschwindigkeiten*,  $\frac{\partial L}{\partial \dot{q}_i} = p_i$  die *generalisierten Impulse* und  $\frac{\partial L}{\partial q_i}$  die *generalisierten Kräfte*.  $S[\vec{q}(t)]$  bezeichnet die *Wirkung*, sie ist eine der zentralen physikalischen Größen. Den von den generalisierten Koordinaten  $\vec{q}$  aufgespannten Raum bezeichnet man als *Konfigurationsraum*, denn jeder Punkt in diesem Raum entspricht einem möglichen Zustand des mechanischen Systems. Den von den Orten  $\vec{q}$  und Impulsen  $\vec{p}$  aufgespannten Raum bezeichnet man als *Phasenraum*. Der Phasenraum hat damit immer eine gerade Dimension.

Das Hamiltonsche Prinzip liefert also die Euler-Lagrange-Gleichungen.

Dem sei ein kurzes Beispiel zur Veranschaulichung zugefügt:

Beispiel: *Harmonischer Oszillator*

Betrachtet wird ein einzelner Massenpunkt, der an einer Feder entlang der  $q$ -Achse schwingt. Die kinetische Energie ist dann durch

$$T = \frac{1}{2} m \dot{q}^2$$

gegeben.

Die potentielle Energie ist die der Feder

$$V = \frac{1}{2}kq^2$$

wobei  $k$  die Hooksche Federkonstante bezeichnet.

Die Lagrange-Funktion ist damit

$$L(q, \dot{q}) = \frac{1}{2}m\dot{q}^2 - \frac{1}{2}kq^2$$

und für die Bewegungsgleichung folgt:

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = \frac{d}{dt} \frac{\partial T}{\partial \dot{q}} - \frac{\partial V}{\partial q} = m\ddot{q} + kq = 0$$

Das ist die bekannte Bewegungsgleichung des harmonischen Oszillators, sie konnte direkt aus dem Hamiltonschen Prinzip heraus aufgestellt werden.

Ein offensichtlicher Nachteil dieses Konzepts ist jedoch der Charakter der Lagrange-Funktion, sie tritt in einem System physikalischer Abläufe nicht als Observable auf und ist für komplexe Situationen nur sehr schwer zu bestimmen.

### 3.1.3 Hamilton-Mechanik

Eine weitere Formulierung der klassischen Mechanik stellt die Hamiltonsche Beschreibung dar. Zusätzlich zu dem sehr allgemeinen Zugang der Lagrange-Mechanik bietet sie neben der besseren Anschaulichkeit der Hamilton-Funktion außerdem noch eine geometrische Form, mit der die Grenzen der klassischen Mechanik ausgeweitet werden können.

#### Kanonische Hamilton-Mechanik

Die zuvor betrachtete Euler-Lagrange-Gleichung (3.4) ist eine Differentialgleichung 2. Ordnung. Je nach Anzahl der Freiheitsgrade  $n$  lässt sie sich komponentenweise in ebenso vielen einzelne Differentialgleichungen anschreiben. Wird das System nun unter Einführung des generalisierten Impulses  $p$  als neue Variable von den Koordinaten  $(q, \dot{q})$  auf  $(q, p)$  transformiert, so ergeben sich  $2n$  Differentialgleichungen 1. Ordnung. Dies wird realisiert durch die Legendre-Transformation und führt zu den Hamiltonschen Gleichungen:

$$\dot{q} = \frac{\partial H}{\partial p} \tag{3.7}$$

$$\dot{p} = -\frac{\partial H}{\partial q} \tag{3.8}$$

Satz: *Hamiltonsche Gleichungen*

Das System der Euler-Lagrange-Gleichungen ist den Hamiltonschen Gleichungen (3.7) und (3.8) äquivalent, wenn

$$H(\vec{q}, \vec{p}, t) = \dot{\vec{p}} \cdot \vec{q} - L(\vec{q}, \dot{\vec{q}}, t) \quad (3.9)$$

die Legendre-Transformation der Lagrange-Funktion  $L$  bezüglich der verallgemeinerten Geschwindigkeiten  $\dot{\vec{q}}$  ist.  $H(\vec{q}, \dot{\vec{q}}, t)$  bezeichnet man als Hamilton-Funktion.

Ein Vorteil dieser Schreibweise wird sofort bei der Betrachtung der physikalischen Bedeutung von  $H$  deutlich. Im Gegensatz zu der Lagrange-Funktion, deren natürliche Form die Differenz aus kinetischer und potentieller Energie ist, berechnet sich die Hamilton-Funktion aus der Summe beider Energien. Das dynamische Verhalten eines Systems kann also durch seine Gesamtenergie, repräsentiert durch die skalare Hamilton-Funktion, beschrieben werden. Insbesondere die Erhaltung dieser Gesamtenergie bei konservativen Systemen, und damit auch der Hamilton-Funktion, eröffnet eine Vielzahl an neuen Möglichkeiten in der Untersuchung von Systemdynamiken.

## Geometrische Formulierung

Um diese Vorteile nutzen zu können wird nun die geometrische Formulierung der Hamilton-Mechanik in den Vordergrund gestellt.

Aus der Herleitung der Hamiltonschen Gleichungen durch die Legendre-Transformation der Euler-Lagrange-Gleichungen folgt zunächst die Einschränkung auf Phasenräume mit gerader Dimension.

Wird die Größe

$$\vec{z} = \begin{pmatrix} \vec{q} \\ \vec{p} \end{pmatrix}$$

zur Beschreibung der Phase eingeführt, so lässt sich das System (3.7), (3.8) in Matrix-Vektor-Form folgendermaßen formulieren:

$$\frac{d\vec{z}}{dt} = \mathbb{J}_c \cdot \frac{\partial H}{\partial \vec{z}} \quad (3.10)$$

wobei

$$\mathbb{J}_c = \begin{pmatrix} 0 & \mathbb{I} \\ -\mathbb{I} & 0 \end{pmatrix} \quad (3.11)$$

als *kanonischer Poissontensor* bezeichnet wird.  $\mathbb{I}$  und  $0$  sind die  $n$ -dimensionale Einheits- bzw. Nullmatrix,  $\mathbb{J}_c$  ist also  $2n$ -dimensional.

### 3 Alternative Formulierungen der klassischen Mechanik

Für die zuvor angesprochene Verallgemeinerung soll nun über das kanonische Hamilton-System die sogenannte *Poissonklammer* eingeführt werden.

Wird eine beliebige Funktion in Phasenraumkoordinaten zeitlich differenziert, so liefert die Kettenregel:

$$\frac{dF(\vec{z})}{dt} = \frac{\partial F}{\partial \vec{q}} \frac{d\vec{q}}{dt} + \frac{\partial F}{\partial \vec{p}} \frac{d\vec{p}}{dt} \quad (3.12)$$

Durch Einsetzen der Hamiltonschen Gleichungen ergibt sich

$$\frac{dF(\vec{z})}{dt} = \frac{\partial F}{\partial \vec{q}} \frac{\partial H}{\partial \vec{p}} - \frac{\partial F}{\partial \vec{p}} \frac{\partial H}{\partial \vec{q}} =: \{F, H\} \quad (3.13)$$

die Definition der kanonischen Poissonklammer  $\{F, H\}$ .

Damit können die kanonischen Gleichungen in symmetrischer Form geschrieben werden:

$$\frac{d\vec{q}}{dt} = \{\vec{q}, H\} \quad (3.14)$$

$$\frac{d\vec{p}}{dt} = \{\vec{p}, H\} \quad (3.15)$$

Die ausschlaggebende Variable in der Definition der Poissonklammer ist der (hier kanonische) Poissontensor  $\mathbb{J}_c$ :

$$\{F, G\} = \begin{pmatrix} \frac{\partial F}{\partial \vec{q}} & \frac{\partial F}{\partial \vec{p}} \end{pmatrix} \begin{pmatrix} 0 & \mathbb{I} \\ -\mathbb{I} & 0 \end{pmatrix} \begin{pmatrix} \frac{\partial G}{\partial \vec{q}} \\ \frac{\partial G}{\partial \vec{p}} \end{pmatrix} \quad (3.16)$$

Es gilt Antisymmetrie:

$$\{F, G\} = -\{G, F\} \quad (3.17)$$

und für 3 beliebige Funktionen  $F, G$  und  $H$  die Jacobi-Identität:

$$\{F, \{G, H\}\} + \{G, \{H, F\}\} + \{H, \{F, G\}\} = 0 \quad (3.18)$$

Die Poissonklammer entspricht einer bilinearen Abbildung.

Nun zum Ertrag all dieser Vorbemerkungen:

Definition: *Hamiltonsches System*



### 3 Alternative Formulierungen der klassischen Mechanik

Ein System heißt Hamiltonsch, wenn es zum einen einen Phasenraum  $M$  und eine Abbildung (genannt Poissonklammer), die je zwei Funktionen  $F, G : M \rightarrow R$  eine weitere Funktion  $\{F, G\}$  zuordnet, gibt, sodass die beiden Eigenschaften (3.17) und (3.18) erfüllt sind.

Zum anderen muss es eine Hamilton-Funktion  $H : M \rightarrow R$  geben, sodass sich die Evolutionsgleichungen des Systems als

$$\frac{d\vec{z}}{dt} = \{\vec{z}, H\} \quad (3.19)$$

schreiben lassen.

Ein Hamiltonsches System besteht also aus zwei grundlegenden Komponenten:

- dem Poissontensor und
- der Hamilton-Funktion.

Der Poissontensor spiegelt die Geometrie des Phasenraums wieder, stellt also den kinematischen Teil des Systems dar.

Die Hamilton-Funktion zeigt die zeitliche Entwicklung des Modellzustandes, ist also als dynamischer Teil anzusehen. [27]

Die Poissonklammer kann auch zur Charakterisierung von Erhaltungsgrößen verwendet werden, denn für jede konservative Funktion  $F(z)$  muss definitionsgemäß gelten:

$$\frac{dF}{dt} = \{F, H\} = 0 \quad (3.20)$$

Aus zwei bekannten Erhaltungsgrößen kann so unter Verwendung der Jacobi-Identität eine dritte konstruiert werden.

Ein Vergleich in [27] zeigt wie naheliegend die Verwendung des Hamilton-Formalismus für meteorologische Zwecke ist:

Der geostrophische Wind in Druckkoordinaten schreibt sich:

$$\vec{v}_g = (u, v) = \vec{k} \times \nabla \Psi \quad (3.21)$$

wobei mit

$$\Psi = \frac{\phi}{f_0}$$

die Stromfunktion gemeint ist, für die gilt

$$\frac{dy}{dt} = \frac{\partial \Psi}{\partial x} \quad \frac{dx}{dt} = -\frac{\partial \Psi}{\partial y}$$

Es lässt sich folgender Vergleich ziehen (Tab. 3.1<sup>1</sup>)

Geostrophischer Wind	Kanonische Dynamik
$\frac{d\vec{x}}{dt} = \vec{k} \times \nabla \Psi$ <p><math>\vec{k}</math> : vertikaler Einheitsvektor senkrecht auf die <math>(x, y)</math> - Ebene im Ortsraum</p>	$\frac{d\vec{x}}{dt} = \vec{k} \times \nabla H$ <p><math>\vec{k}</math> : vertikaler Einheitsvektor senkrecht auf die <math>(p, q)</math> - Ebene im Phasenraum</p>
<p>Ortsraum</p> $\vec{x} = (x, y)$	<p>Phasenraum</p> $\vec{x} = (p, q)$
<p>Geopotential als Stromfunktion im Ortsraum, bestimmt die gesamte “synoptische” Dynamik</p>	<p>Hamilton-Funktion oder Gesamtenergie bestimmt als Stromfunktion die Dynamik im Phasenraum</p>

Tabelle 3.1: Vergleich: Geostrophischer Wind - Kanonische Dynamik

### Nicht kanonische Hamilton Mechanik

An die Poissonklammer werden nur die beiden oben beschriebenen Forderungen der Antisymmetrie und der Gültigkeit der Jacobi-Identität gestellt, zu keinem Zeitpunkt wurde festgelegt, dass  $\det \mathbb{J} \neq 0$ .

Ist dies der Fall, so ist der Poissontensor kanonisch oder kann auf solch einen transformiert werden.

Um sich jedoch von der starren kanonischen Denkweise zu lösen, wird nun auch eine Singularität bezüglich  $\mathbb{J}$  zugelassen. Dies führt zu einer universelleren Betrachtungsweise dynamischer Systeme und der Auflösung zahlreicher Einschränkungen.

---

<sup>1</sup>Quelle: [27]

### 3 Alternative Formulierungen der klassischen Mechanik

Wird eine Matrix als singulär angenommen, so gibt es Vektoren abseits der trivialen Nulllösung, die im Nullraum der Matrix liegen.

$$\exists \vec{x} \neq 0 : \mathbb{J} \vec{x} = 0 \quad (3.22)$$

Anders gesagt: es gibt nicht verschwindende Eigenvektoren zum Null-Eigenwert, deren Anzahl durch die Dimension  $n$  des Nullraums gegeben ist.

Es kann gezeigt werden, dass diese Eigenvektoren eines singulären Poissontensor immer als Gradient skalarer Funktionen  $C^i$ ,  $i = 1, \dots, n$  darstellbar sind:

$$\mathbb{J} \frac{\partial C^i}{\partial \vec{x}} = 0 \quad (3.23)$$

Die Funktionen  $C^i$ ,  $i = 1, \dots, n$  werden *Casimir-Funktionen* des Poissontensors genannt. Da (3.23) gilt, kann für jede beliebige Funktion  $F$  geschrieben werden

$$\{F, C^i\} = \left( \frac{\partial F}{\partial \vec{x}} \right)^T \mathbb{J} \frac{\partial C^i}{\partial \vec{x}} = 0 \quad (3.24)$$

und insbesondere natürlich auch

$$\frac{dC^i}{dt} = \{C^i, H\} = 0 \quad (3.25)$$

Jede Casimir-Funktion ist somit eine Erhaltungsgröße für sein Hamiltonsches System und maßgeblich für das dynamische Verhalten.

In [3] sehr treffend formuliert:

“Geometrisch bedeutet die Existenz solcher ausgezeichneten Funktionen, dass die Systementwicklung nicht frei ist, denn die Erhaltung der Casimire führt dazu, dass das System die Flächen  $C^1 = \text{const}, \dots, C^n = \text{const}$  nicht verlassen darf. Die Bewegung findet dann auf der Schnittfläche dieser Casimire statt und man kann zeigen, dass auf dieser Schnittfläche wieder kanonische Koordinaten existieren. Casimire sind also kinematische Constraints, die *unabhängig von der jeweiligen Hamilton-Funktion* in den Phasenraum eingebaut sind.”

Dieses Konzept ist in der atmosphärischen Physik von großer Bedeutung, da die Transformation von Lagrangeschen (Teilchenkoordinaten, Zeit) auf Eulersche Koordinaten (Raumkoordinaten, Zeit) nicht kanonisch ist und somit die Hamilton-Darstellung in Eulerschen Koordinaten singulär ist.

### 3.1.4 Nambu-Mechanik

Literatur: [7, 27]

Die Nambu Mechanik stellt eine von dem Japaner YOICHIRO NAMBU in den 1970ern entwickelte Verallgemeinerung der Hamiltonschen Mechanik dar.

Es stellte sich unter anderem die Frage, warum trotz Auftreten zweier oder mehrerer der Hamilton-Funktion gleichwertiger Erhaltungsgrößen (Casimire) bei singulären Poissonensoren, nur  $H$  zur Beschreibung der Systemdynamik verwendet werden kann. Die Existenz dieser weiteren Erhaltungsgrößen wird somit bei der mathematischen Behandlung des Modells nicht berücksichtigt, obwohl diese für das Systemverhalten von höchster Relevanz sind.

So wurde das Konzept der bilinearen Poissonklammer auf die multilineare Nambuklammer erweitert, mit Hilfe derer mehrere Erhaltungsgrößen in die Beschreibung der Systementwicklung mit einbezogen werden können. Die zeitliche Entwicklung hängt also nicht mehr nur von der Hamilton-Funktion ab, sondern berücksichtigt auch andere modellspezifische Erhaltungsgrößen.

Die Nambuklammer ist, wie die Poissonklammer, antisymmetrisch und erfüllt eine Verallgemeinerung der Jacobi-Identität.

Bei Betrachtung eines  $n$  - dimensionalen, punktmechanischen Systems mit  $n - 1$  voneinander unabhängigen Erhaltungsgrößen  $H_j$ ,  $j = 1, \dots, n - 1$ , lässt sich das Differentialgleichungssystem zur zeitlichen Entwicklung des Modellzustands in folgender Form angeben:

$$\frac{dF}{dt} = \frac{\partial (F, H_1, \dots, H_{n-1})}{\partial (z_1, z_2, \dots, z_n)} =: \{F, H_1, H_2, \dots, H_{n-1}\} \quad (3.26)$$

Es wurden diskrete Modelle gefunden, die eine Formulierung nach Nambu zulassen, unter anderem auch das Lorenz(1963)-Modell. Allerdings konnten in der Praxis bis jetzt keine Modelle mit mehr als einer zusätzlichen Erhaltungsgröße formuliert werden. Diese Einschränkung der Nambuklammer hängt wahrscheinlich mit der zugrunde liegenden Lie Algebra zusammen, auf der schon die Poissonklammer basiert. Zwar scheint der Erfolg der Verwendung einer einzigen zusätzlichen Casimir-Funktion bei einer unendlichen Anzahl an Freiheitsgraden bei partiellen Differentialgleichungssystemen zunächst eher bescheiden, jedoch bringt auch diese Erweiterung schon signifikante Vorteile gegenüber der klassischen Hamiltonschen Mechanik mit sich. Bei der Behandlung kontinuierlicher Problemstellungen wie Feldgleichungen stellte sich heraus, dass durch die Integration einer zusätzlichen Erhaltungsgröße in den Modellgleichungen, also durch die Einführung einer trilinearen Nambuklammer anstelle der bilinearen Poissonklammer, Singularitäten

formal beseitigt werden können.

Der fixe Zusammenhang zwischen der Dimensionalität des Phasenraumes und der Anzahl der Erhaltungsgrößen kann in der kontinuierlichen Nambu-Mechanik vermieden werden. In der atmosphärischen Physik wird die Verallgemeinerung der Nambu-Mechanik *energy-vorticity-theory* (*Energie-Wirbel-Theorie*) genannt, da die zusätzlich verwendeten Casimir-Funktionen meist integrale Größen der Vorticity darstellen.

Wie bereits erwähnt steht der Poissontensor bzw. die daraus resultierende Poissonklammer für die Geometrie des physikalischen Modells im Phasenraum. Da das System seine konservativen Eigenschaften nicht verletzen darf, kann sich die Trajektorie im Phasenraum nur auf Flächen bewegen, auf denen die Konstanz der Casimire gewährleistet ist. Verfügen die Entwicklungsgleichungen nun über mehrere Erhaltungsgrößen, bleiben für die möglichen Modellzustände nur noch die Schnittflächen aller dieser Größen übrig. Zuzufolge dieser Erkenntnis kann von den Modellgleichungen direkt auf Aussehen und Eigenschaften des entstehenden Attraktor geschlossen werden.

## 3.2 Betrachtung des Lorenz(1963)-Modells in Hamilton- bzw. Nambu-Mechanik

### 3.2.1 Hamiltonsche Formulierung der Lorenz-Gleichungen

Zur Hamiltonschen Formulierung wird nur der konservative Kern des Lorenz-Gleichungssystems (2.38) - (2.40)

$$\frac{dX}{dt} = \sigma Y \quad (3.27)$$

$$\frac{dY}{dt} = -XZ + rX \quad (3.28)$$

$$\frac{dZ}{dt} = XY \quad (3.29)$$

betrachtet, in ihm steckt der gesamte nichtlineare Charakter der zugrunde liegenden Konvektionsgleichungen.

Zunächst muss eine geeignete Hamilton-Funktion, also eine zeitinvariante Größe für das System gefunden werden. Dazu wird das Verhältnis von (3.28) und (3.29) gebildet:

$$\frac{\frac{dY}{dt}}{\frac{dZ}{dt}} = \frac{dY}{dZ} = \frac{r - Z}{Y}$$

### 3 Alternative Formulierungen der klassischen Mechanik

Werden die Variablen separiert

$$Y dy = (r - Z) dz$$

und die Integration durchgeführt, so resultiert die Erhaltungsgröße

$$\mathcal{H} = \frac{1}{2}Y^2 + \frac{1}{2}Z^2 - rZ = \text{const} \quad (3.30)$$

Weiters wird der antisymmetrische Tensor gesucht, der für

$$\vec{x} = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

die Beziehung

$$\frac{dx_i}{dt} = \mathbb{J}_{ij} \frac{\partial \mathcal{H}}{\partial x_j} \quad (3.31)$$

erfüllt.

In diesem Fall ist es leicht zu sehen, dass

$$\mathbb{J} = \begin{pmatrix} 0 & \sigma & 0 \\ -\sigma & 0 & -X \\ 0 & X & 0 \end{pmatrix}$$

sein muss.

Es wird sofort klar, dass das konservative Lorenz(1963)-Modell tatsächlich als nichtkanonisches Hamiltonsches System formuliert werden kann, dessen Poissontensor  $\mathbb{J}$  singular ist, da  $\det(\mathbb{J}) = 0$ .

Der Tensor  $\mathbb{J}$  verfügt über zwei nicht linear abhängige Spalten, es muss also noch eine zweite Erhaltungsgröße, eine Casimir-Funktion  $\mathcal{C}$  existieren, welche durch Lösung der Gleichung

$$\begin{pmatrix} 0 & \sigma & 0 \\ -\sigma & 0 & -X \\ 0 & X & 0 \end{pmatrix} \begin{pmatrix} \frac{\partial \mathcal{C}}{\partial x} \\ \frac{\partial \mathcal{C}}{\partial y} \\ \frac{\partial \mathcal{C}}{\partial z} \end{pmatrix} = 0 \quad (3.32)$$

angeschrieben werden kann als

$$\mathcal{C} = \frac{1}{2}X^2 - \sigma Z \quad (3.33)$$

Die Hamiltonsche Darstellung lautet:

$\frac{d\vec{x}}{dt} = \{\vec{x}, \mathcal{H}\} \quad (3.34)$
---

Es tritt der in Kap. 3.1.4 beschriebene Fall auf, dass es im Modell zwei gleichwertige Erhaltungsgrößen gibt, allerdings nur eine davon für die Hamiltonsche Beschreibung herangezogen werden kann.

Darum wird nun zur Nambu-mechanischen Darstellung des Modells übergegangen um zu zeigen, dass eine Einbeziehung beider relevanten Größen sehr wohl möglich ist und auch zu einer sehr schönen geometrischen Interpretation führt.

### 3.2.2 Nambu-mechanische Formulierung der Lorenz-Gleichungen

Besteht der Vorteil einer Hamilton-Formulierung in der Garantie der Erhaltung der Gesamtenergie, so geht die Nambu-Mechanik mit einer trilinearen Klammerstruktur noch einen Schritt weiter und versichert auch die Erhaltung einer zweiten Größe.

Gesucht wird also eine Formulierung der Evolutionsgleichung des konservativen Lorenz Modell (3.27) bis (3.29) in der Art von

$$\frac{dx_i}{dt} = \frac{\partial (x_i, H_1, H_2)}{\partial (x_1, x_2, x_3)} \quad (3.35)$$

mit

$$H_1 = \mathcal{H} = \frac{1}{2}Y^2 + \frac{1}{2}Z^2 - rZ \quad (3.36)$$

und

$$H_2 = \mathcal{C} = \frac{1}{2}X^2 - \sigma Z \quad (3.37)$$

(3.35) komponentenweise:

$$\frac{dX}{dt} = \frac{\partial (H_1, H_2)}{\partial (Y, Z)} = \sigma Y$$

$$\frac{dY}{dt} = \frac{\partial (H_1, H_2)}{\partial (Z, X)} = rX - XZ$$

$$\frac{dZ}{dt} = \frac{\partial (H_1, H_2)}{\partial (X, Y)} = XY$$

Der konservative Teil des Lorenz(1963)-Modell kann tatsächlich als Nambu-mechanisches Gleichungssystem aufgefasst werden, in dem beide Erhaltungsgrößen explizit vorkommen. Dies und darauf folgende Betrachtungen werden zum ersten Mal von P. N  vir und R. Blender 1994 in [28] gezeigt:

$$\frac{d\vec{x}}{dt} = \nabla \mathcal{H} \times \nabla \mathcal{C} \quad (3.38)$$

Da die Gr   en  $\mathcal{H}$  und  $\mathcal{C}$  Zwangsbedingungen f  r die zeitliche Entwicklung des Modellzustandes darstellen, l  sst sich der Attraktor im Phasenraum direkt aus der Evolutionsgleichung (3.38) konstruieren, indem die Schnittlinie der beiden geometrischen Objekte  $\mathcal{H}$  und  $\mathcal{C}$  gebildet wird - auf dieser kann sich die Trajektorie w  hrend der Modellintegration bewegen, siehe Abb. 3.1<sup>2</sup>. Die Geschwindigkeit wird festgelegt durch die Gradienten und Schnittwinkel der beiden Fl  chen der Erhaltungsgr   en.

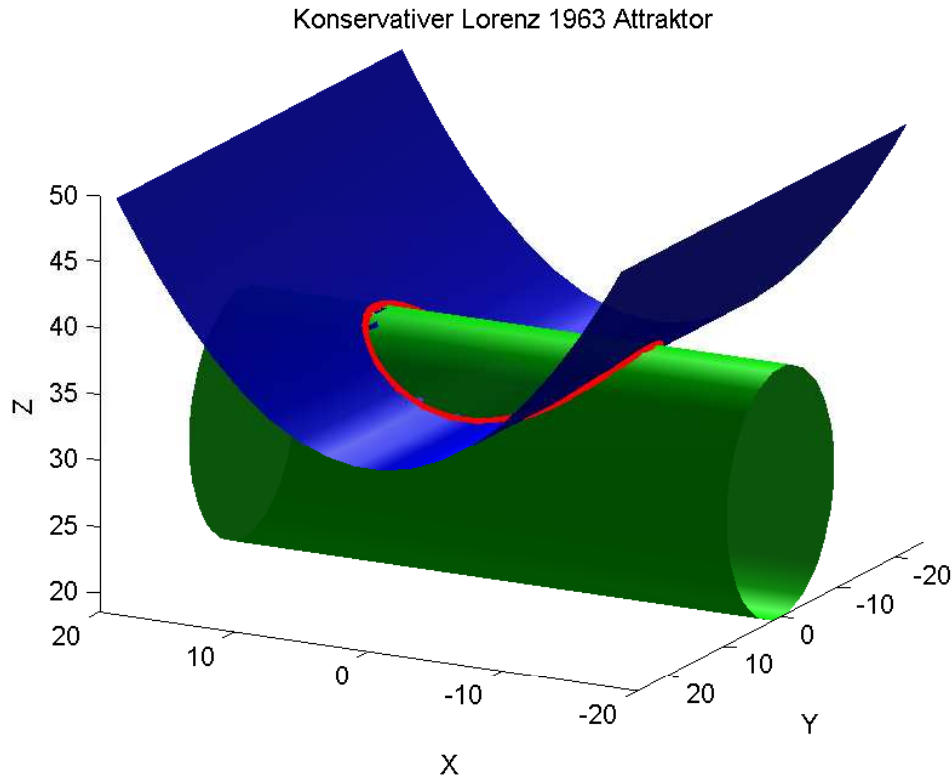


Abbildung 3.1: Geometrische Interpretation des Lorenz(1963)-Attraktors als Schnittmenge der beiden Erhaltungsgr   en  $H$  und  $C$

Nicht nur jeder Casimir alleine, sondern auch alle Linearkombinationen aus diesen stellen

<sup>2</sup>Quelle: Code auf Seite 173



### 3 Alternative Formulierungen der klassischen Mechanik

wieder Erhaltungsgrößen dar. Somit kann die relative Position der beiden Objekte der geometrischen Darstellung variiert werden und verschiedene dynamische Aspekte können durch speziell gewählte Parameterkonfigurationen im Anfangszustand untersucht werden. Die in dieser Arbeit verwendete Form des Attraktors ist demnach keineswegs die einzig mögliche.

Dass das System tatsächlich konservativ ist, zeigt die Divergenz

$$\nabla \cdot \frac{d\vec{x}}{dt} = \nabla \cdot (\nabla \mathcal{H} \times \nabla \mathcal{C}) = 0 \quad (3.39)$$

Zur Berücksichtigung der Dissipation wird in [28] der Parameter  $m$  eingeführt, über den sich die relative Größe der zusätzlichen Terme steuern lässt.

$$\frac{dX}{dt} = -m\sigma X + \sigma Y \quad (3.40)$$

$$\frac{dY}{dt} = -XZ + rX - mY \quad (3.41)$$

$$\frac{dZ}{dt} = XY - m b Z \quad (3.42)$$

Das System lässt sich dann in seiner vollständigen Form schreiben als:

$$\frac{d\vec{x}}{dt} = \nabla \mathcal{H} \times \nabla \mathcal{C} - \nabla \Phi \quad (3.43)$$

wobei  $\Phi$  die Funktion der Dissipation

$$\Phi = \frac{m}{2} (\sigma X^2 + Y^2 + b Z^2) \quad (3.44)$$

darstellt.

Für eine beliebige Funktion  $F(\vec{x})$  wird durch

$$\frac{dF(\vec{x})}{dt} = \nabla F \cdot (\nabla \mathcal{H} \times \nabla \mathcal{C}) - \nabla F \cdot \nabla \Phi \quad (3.45)$$

deutlich, warum die geometrische Interpretation des dissipativen Systems sich als schwierig erweist: die Flächen der Erhaltungsgrößen und somit auch ihre Schnittlinie sind nicht mehr zeitinvariant.

Genauere Betrachtungen hierzu können zudem in Peter Névirs Habilitationsschrift [26] aus dem Jahr 1998 gefunden werden.

Eine interessante Darstellungsmöglichkeit des dissipativen Systems wurde außerdem 2008 in Nambu-metriplektischer Form in [4] hergeleitet.

### 3.2.3 Problematik der Lorenz-Gleichungen

Einer der Hauptgründe für die Einführung alternativer Herangehensweisen zur Beschreibung des Lorenz(1963)-Modells in diesem Kapitel ist die Möglichkeit unter Verwendung dieser Formalismen gravierende Unstimmigkeiten im Konzept des finite-mode models aufzuzeigen.

Die Schwierigkeit in der Entwicklung solch eines reduzierten Gleichungssystems liegt ganz offensichtlich in der Entscheidung, welche Moden für eine adäquate Beschreibung des physikalischen Systems beibehalten werden müssen. Da es keine patentierte Vorgehensweise oder allgemein gültige Kriterien zur Herleitung gibt, stellt sich im Nachhinein immer noch die Frage, ob das System endlicher Differentialgleichungen wirklich die Struktur und die Charakteristik der ursprünglich partiellen Differentialgleichungen widerspiegelt.

Aus der Sicht der Hamiltonschen Mechanik kann gesagt werden, dass die Erhaltung der Hamiltonschen Struktur der Originalgleichungen unumgänglich ist. Aus dieser Überprüfung sollen nun einige Eckpunkte skizziert werden.

Zwecks Untersuchung der Strukturerhaltung wird die kontinuierliche Poissonklammer der physikalischen Gleichungen mit der diskreten Poissonklammer des spektralen Modells verglichen.

Für die Konvektionsgleichungen

$$\frac{\partial \zeta}{\partial t} + J(\psi, \zeta) - R\sigma \frac{\partial T}{\partial x} = \sigma \nabla^4 \psi \quad (3.46)$$

und

$$\frac{\partial T}{\partial t} + J(\psi, T) - \frac{\partial \psi}{\partial x} = \nabla^2 T \quad (3.47)$$

muss durch Probieren und mit Hilfe von etwas Erfahrung die Hamiltonsche Form ermittelt werden.

Die nichtkanonische Poissonklammer muss dann

$$\{\mathcal{F}, \mathcal{G}\} = \int_F \left( \zeta J \left( \frac{\delta \mathcal{F}}{\delta \zeta}, \frac{\delta \mathcal{G}}{\delta \zeta} \right) + (T - z) \left[ J \left( \frac{\delta \mathcal{F}}{\delta \zeta}, \frac{\delta \mathcal{G}}{\delta T} \right) + J \left( \frac{\delta \mathcal{F}}{\delta T}, \frac{\delta \mathcal{G}}{\delta \zeta} \right) \right] \right) df \quad (3.48)$$

lauten. Die Ermittlung dieser Klammer ist nicht ganz trivial, jedoch kann in [25] eine

### 3 Alternative Formulierungen der klassischen Mechanik

analoge Herleitung für ein magnetohydrodynamisches System nachgelesen werden. Dieses verfügt über eine gleiche Klammerstruktur, jedoch haben die physikalischen Variablen andere Bedeutungen.

So kann das System mittels

$$\frac{d\vec{x}}{dt} = \{\vec{x}, \mathcal{H}\} \quad (3.49)$$

dargestellt werden, wobei die Hamilton-Funktion, also die Summe von kinetischer und potentieller Energie folgendermaßen lautet:

$$\mathcal{H} = \int_F \left( \frac{1}{2} (\nabla \psi)^2 - R\sigma T z \right) df \quad (3.50)$$

Weitere Casimire sind

$$\mathcal{C} = \int_F \zeta h_{(T-z)} df \quad (3.51)$$

und

$$\mathcal{S} = \int_F g_{(T-z)} df \quad (3.52)$$

wobei  $g$  und  $h$  beliebige Funktionen von  $(T - z)$  sind.

Physikalisch interpretiert kann  $\mathcal{C}$  zurückgeführt werden auf Kelvins Zirkulationstheorem, in diesem Fall bleibt die Vorticity entlang einer  $(T - z)$ -Kontur erhalten. Größen der Form  $\mathcal{S}$  hängen mit der Erhaltung der  $(T - z)$ -Konturen zusammen und treten in divergenzfreien, reibungsfreien Fluidmodellen auf.

Nun wird die Struktur der Originalgleichungen mit der des reduzierten Systems verglichen. Schon an der Poissonklammer (3.48) ist zu erkennen, dass daraus der Tensor der Hamiltonschen Darstellung

$$\mathbb{J} = \begin{pmatrix} 0 & \sigma & 0 \\ -\sigma & 0 & -X \\ 0 & X & 0 \end{pmatrix}$$

des Lorenz(1963)-Modells nicht abgeleitet werden kann.

Die Struktur der kontinuierlichen Poissonklammer ist aber eng verbunden mit den physikalischen Abläufen im betrachteten Modell. Klammern wie (3.48) deuten auf Advektion einer Größe durch eine andere hin, was in den Rayleigh-Bénard-Gleichungen auf jeden Fall realisiert wird, wenn Temperaturabweichungen von der Stromfunktion advehiert werden.

### 3 Alternative Formulierungen der klassischen Mechanik

Das Lorenz(1963)-Modell kann dies nicht in seine Systemdynamik integrieren, dazu müsste die Klammer folgende Form haben

$$\{F, G\} = -\vec{\Pi} \cdot (\nabla_{\vec{\Pi}} F \times \nabla_{\vec{\Pi}} G) - \vec{\Gamma} \cdot (\nabla_{\vec{\Pi}} F \times \nabla_{\vec{\Gamma}} G - \nabla_{\vec{\Pi}} G \times \nabla_{\vec{\Gamma}} F) \quad (3.53)$$

wobei  $\vec{\Pi}$  der Drehimpulsvektor und  $\vec{\Gamma}$  der Richtungsvektor der Gravitation aus der Sicht des Systems selbst sind. Diese Klammer der Konvektionsgleichungen stellt strukturell eine „semidirekte Erweiterung“ der Vorticityklammer für das Lorenz(1960)-Modell (siehe z.B. [3]) dar. Beschreibt diese die Dynamik eines *starren Körpers*, so kann 3.53 als Erweiterung auf einen *schweren starren Körper* interpretiert werden, die zusätzlich die ausschlaggebenden Konvektionmechanismen erfasst.

Da

$$\vec{\Pi}, \vec{\Gamma} \in \mathbb{R}^3$$

wären dazu sechs statt drei Dimensionen, also Fourierkomponenten vor der Abschneidung, notwendig. Somit ist das Lorenz(1963)-Modell kein strukturerhaltendes finite-mode model, das die physikalische Komplexität der Rayleigh-Bénard-Gleichungen erfassen und wiedergeben kann.

Auch im Formalismus der Nambu Mechanik können Gegenbeweise für die Vollständigkeit des reduzierten Systems angegeben werden.

In der Nambu Form kommt sowohl  $\mathcal{H}_1$  als auch  $\mathcal{H}_2$  explizit vor, diese sind allerdings beide auf (3.50) zurückzuführen und beinhalten beide Energieerhaltung. Das Fehlen von Größen in Form von  $\mathcal{S}$  lässt sich durch die Vernachlässigung des dissipativen Anteils erklären. Casimire in Form von  $\mathcal{C}$  sind allerdings bei der Reduktion der Moden verloren gegangen, die spektrale Entwicklung von  $C$  unter dem Lorenz Ansatz ergibt Null.

Somit können auch hier die kontinuierliche und die diskrete Poissonklammer nicht als äquivalent angesehen werden.

Vorgreifend sei noch gesagt, dass die Forderung nach der Erhaltung von Punktsymmetrien zudem noch zeigt, dass hier die gewählten Randbedingungen verletzt werden würden.

Ein 6-dimensionales Modell, dass den Anforderungen der Strukturerhaltung gerecht wird, wurde 2010 von A. Bihlo und J. Staufer in [7] abgeleitet und soll im weiteren Verlauf der Arbeit näher untersucht bzw. implementiert werden.

# 4 Erweiterung des Lorenz(1963)-Modells auf 6 Dimensionen

## 4.1 Herleitung

Folgende Herleitung stammt direkt aus [7]. Allerdings wird in der hier folgenden Zusammenfassung eher die physikalische Bedeutung der einzelnen Schritte als die mathematische Vollständigkeit vordergründig sein.

Die Ausgangsgleichungen:

$$\frac{\partial \zeta}{\partial t} + [\Psi, \zeta] = R\sigma \frac{\partial T}{\partial x} + \sigma \nabla^2 \zeta \quad (4.1)$$

$$\frac{\partial T}{\partial t} + [\Psi, T] = \frac{\partial \Psi}{\partial x} + \nabla^2 T \quad (4.2)$$

Die Definition der einzelnen Terme bleibt wie schon in früheren Kapiteln beschrieben:  $\Psi$  ist die 2-dimensionale Stromfunktion, die divergenzfreie Strömung in  $x - z$ -Richtung generiert,  $\zeta = \nabla^2 \Psi$  ist die Vorticity,  $T$  ist die Temperaturabweichung von einem linearen Temperaturprofil im konduktiven Gleichgewicht,  $\sigma$  ist die Prandtl-Zahl,  $R$  die Rayleigh-Zahl und  $[a, b] = \frac{\partial a}{\partial x} \frac{\partial b}{\partial z} - \frac{\partial a}{\partial z} \frac{\partial b}{\partial x}$  ist der Jacobioperator.

Wiederum soll nur der konservative Kern betrachtet werden, die jeweils letzten Terme  $\sigma \nabla^2 \zeta$  und  $\nabla^2 T$  werden also vernachlässigt. Der betrachtete Raum des realen Modells soll

$$\Omega = [-L, L] \times [0, 1] \quad (4.3)$$

sein.

Das gesuchte 6-dimensionale System soll eine Erweiterung des Lorenz(1963)-Modells sein, soll dieses also enthalten.

Dazu werden die gleichen Randbedingungen wie in Kap. 2.1.4 verwendet:

$$\Psi|_{z=0} = \Psi|_{z=1} = 0 \quad (4.4)$$

$$\Theta|_{z=0} = \Theta|_{z=1} = 0 \quad (4.5)$$

und daraus folgend

$$\zeta|_{z=0} = \zeta|_{z=1} = 0 \quad (4.6)$$

Die Entscheidung über die Beibehaltung bestimmter Moden wird nun nicht aufgrund numerischer Ergebnisse getroffen sondern soll etwas mehr physikalischen Hintergrund zeigen. Das dafür verwendete mächtige mathematische Konzept ist das der *Symmetrien*.

## Symmetrien

Literatur: [30]

Weiters: Dr. Alexander Bihlo

Der Ausdruck Symmetrie erzeugt bei jedermann intuitiv schon eine Vorstellung, deren Erklärung auch ohne Ausdrücke wie Transformationsinvarianz auskommt. Herunter gebrochen auf ein einfaches geometrisches Objekt bewirkt die Symmetrie bei zum Beispiel einem homogenen Zylinder, dass er nach einer Drehung um einen beliebigen Winkel entlang der Längsachse genauso aussieht wie zuvor und die Drehung im Nachhinein nie wieder nachgewiesen werden kann.

Allgemein sind Symmetrien Transformation von Objekten, die das Objekt auf sich selbst abbilden, es also invariant lassen. Im hier betrachteten Fall soll das Objekt aus den Konvektionsgleichungen (4.1) und (4.2) bestehen, die Transformationen sind Punkttransformationen der unabhängigen und abhängigen Variable, also  $t, x, z$  und  $\Psi, T$ .

Es gibt zwei grundlegende Arten von Punktsymmetrien, einerseits kontinuierliche und andererseits diskrete. Ein Beispiel für eine kontinuierliche Symmetrie ist die Rotation des erwähnten Zylinders, die man im Prinzip („kontinuierlich“) in einzelne Rotationen um unendlich kleine Winkel aufteilen kann. Eine Spiegelung hingegen ist per Definition nicht in elementarere Transformationen einteilbar, d.h. eine Spiegelung ist ein Beispiel für eine diskrete Transformation.

Wie zuvor bereits erwähnt sind die Symmetrien von einem Differentialgleichungssystem Transformationen, die das System auf sich selbst abbilden. Eine alternative und vollständig gleichwertige Definition ist, dass Symmetrien Transformationen sind, die Lösungen

eines Differentialgleichungssystems auf andere Lösungen desselben Systems abbilden. Diese zweite Definition unterstreicht sogleich die Nützlichkeit von Symmetrien: Sie können verwendet werden, um aus bekannten Lösungen neue Lösungen zu konstruieren. Ein weiterer großer Nutzen hierbei ist nun, dass auch wenn die exakte Lösung des Systems nicht möglich ist, was häufig bei partiellen Differentialgleichungen der Fall ist, schon aus der Kenntnis der Symmetrien sehr viel über das zugrunde liegende physikalische Verhalten der Differentialgleichungen gelernt werden kann.

Bei der Herleitung des niedrig-dimensionalen Konvektionsmodells können Symmetriekenntnisse als Entscheidungsgrundlage für die Wahl der Moden verwendet werden. Da die Symmetrien der Originalgleichungen erhalten bleiben müssen, können einige Fourierkoeffizienten sofort ausgeschlossen werden, da sie gewisse Invarianzen verletzen würden. Diese Vorgehensweise soll nun [7] folgend skizziert werden:

In einem ersten Schritt werden die kontinuierlichen (oder Lie) Symmetrien der Konvektionsgleichungen (4.1) und (4.2) berechnet. Die Berechnung von kontinuierlichen Symmetrien ist vollständig algorithmisch und kann heutzutage gut mittels Computeralgebrasystemen (z.B. Maple oder Mathematica) bewerkstelligt werden. Diese Berechnung liefert die sogenannten infinitesimalen Erzeugenden der kontinuierlichen Symmetrien. Diese können durch partielle Ableitungsoperatoren dargestellt werden (siehe [30] für weitere Details). Für die Konvektionsgleichungen (4.1) und (4.2) lauten die infinitesimalen Generatoren

$$\begin{aligned}\mathcal{D} &= 2t\partial_t + x\partial_x + z\partial_z - (3T - 4Rz)\partial_T, \quad \partial_t, \quad \partial_z, \\ \mathcal{X}_1(f) &= f(t)\partial_z + f(t)R\partial_T + f'(t)x\partial_\Psi, \quad \mathcal{X}_2(h) = h(t)\partial_x - h'(t)z\partial_\Psi, \\ \mathcal{Z}(g) &= g(t)\partial_\Psi\end{aligned}\tag{4.7}$$

wobei  $f, g$  und  $h$  beliebige Funktionen der Zeit  $t$  sind.

Um die physikalische Bedeutung dieser infinitesimalen Symmetrietransformationen deutlicher zu machen ist es hilfreich, die endliche Form der obigen Transformationen zu berechnen.

Für einen infinitesimalen Generator der Form

$$Q = \tau(t, x, z, \Psi, T)\partial_t + \xi(t, x, z, \Psi, T)\partial_x + \eta(t, x, z, \Psi, T)\partial_z + \phi(t, x, z, \Psi, T)\partial_\Psi + \theta(t, x, z, \Psi, T)\partial_T\tag{4.8}$$

geschieht dies allgemein durch Lösung des folgenden Systems von gewöhnlichen Differentialgleichungen erster Ordnung (siehe erneut [30] für weitere Details):

$$\frac{d\tilde{t}}{d\varepsilon} = \tau(\tilde{t}, \tilde{x}, \tilde{z}, \tilde{\Psi}, \tilde{T})$$

$$\frac{d\tilde{x}}{d\varepsilon} = \xi(\tilde{t}, \tilde{x}, \tilde{z}, \tilde{\Psi}, \tilde{T})$$

$$\frac{d\tilde{z}}{d\varepsilon} = \eta(\tilde{t}, \tilde{x}, \tilde{z}, \tilde{\Psi}, \tilde{T})$$

$$\frac{d\tilde{\Psi}}{d\varepsilon} = \phi(\tilde{t}, \tilde{x}, \tilde{z}, \tilde{\Psi}, \tilde{T})$$

$$\frac{d\tilde{T}}{d\varepsilon} = \theta(\tilde{t}, \tilde{x}, \tilde{z}, \tilde{\Psi}, \tilde{T})$$

mit den Anfangsbedingungen  $\tilde{t}(\varepsilon = 0) = t$ ,  $\tilde{x}(\varepsilon = 0) = x$ ,  $\tilde{z}(\varepsilon = 0) = z$ ,  $\tilde{\Psi}(\varepsilon = 0) = \Psi$ ,  $\tilde{T}(\varepsilon = 0) = T$ , d.h. die identische Transformation entspricht dem Parameterwert  $\varepsilon = 0$ . Meteorologisch interpretiert ist das Lösen jenes Gleichungssystems analog zur Rekonstruktion der Trajektorie eines Partikels wenn das Geschwindigkeitsfeld bekannt ist.

Für die Generatoren (4.7) der Konvektionsgleichungen ergibt sich durch Lösen dieses Systems:

$$\Gamma_1: \quad (\tilde{t}, \tilde{x}, \tilde{z}, \tilde{\Psi}, \tilde{T}) = (e^{2\varepsilon_1}t, e^{\varepsilon_1}x, e^{\varepsilon_1}z, \Psi, e^{-3\varepsilon_1}(T - Rz) + e^{\varepsilon_1}Rz) \quad (4.9)$$

$$\Gamma_2: \quad (\tilde{t}, \tilde{x}, \tilde{z}, \tilde{\Psi}, \tilde{T}) = (t + \varepsilon_2, x, z, \Psi, T) \quad (4.10)$$

$$\Gamma_3: \quad (\tilde{t}, \tilde{x}, \tilde{z}, \tilde{\Psi}, \tilde{T}) = (t, x, z + \varepsilon_3, \Psi, T) \quad (4.11)$$

$$\Gamma_4: \quad (\tilde{t}, \tilde{x}, \tilde{z}, \tilde{\Psi}, \tilde{T}) = (t, x, z + f(t)\varepsilon_4, \Psi + f'(t)\varepsilon_4x, T + Rf(t)\varepsilon_4) \quad (4.12)$$

$$\Gamma_5: \quad (\tilde{t}, \tilde{x}, \tilde{z}, \tilde{\Psi}, \tilde{T}) = (t, x + h(t)\varepsilon_5, z, \Psi - h'(t)\varepsilon_5z, T) \quad (4.13)$$

$$\Gamma_6: \quad (\tilde{t}, \tilde{x}, \tilde{z}, \tilde{\Psi}, \tilde{T}) = (t, x, z, \Psi + g(t)\varepsilon_6, T) \quad (4.14)$$

wobei  $\varepsilon_1, \dots, \varepsilon_6 \in \mathbb{R}$ .

Physikalisch stellen diese kontinuierlichen Transformationen  $\Gamma_1$  -  $\Gamma_6$  also (4.9) Skalierungen, (4.10) und (4.11) Translationen in Zeit- und  $z$ -Richtung, (4.12) und (4.13) verallgemeinerte Galilei-Transformationen in  $x$ - und  $z$ -Richtung, sowie (4.14) Eichung der Stromfunktion mit einer beliebigen zeitabhängigen Funktion dar.

Außerdem können durch einfaches Probieren zwei unabhängige diskrete Symmetrien der Gleichungen (4.1)–(4.2) ermittelt werden.

Diese sind

$$e_1: (t, x, z, \Psi, T) \mapsto (t, x, -z, -\Psi, -T) \quad (4.15)$$



und

$$e_2 : (t, x, z, \Psi, T) \mapsto (t, -x, z, -\Psi, T) \quad (4.16)$$

Die oben berechneten Punktsymmetrien gelten allerdings nur für die freien Konvektionsgleichungen, d.h. der Existenz von Randbedingungen wurde noch nicht Rechnung getragen. Die gemeinsame Betrachtung mit Randbedingungen reduziert die Symmetrieeigenschaften des Systems zwar drastisch, ist aber gerade beim Aufbau der Rayleigh-Bénard-Konvektion unumgänglich.

Die Untersuchung dieses Problems bringt folgende Erkenntnisse:

Nachdem die zugehörige Domäne  $\Omega = [-L, L] \times [0, 1]$  fest ist, kann keine Skalierungssymmetrie angewendet werden, da dieser gegenüber die Domäne ja nicht invariant wäre. Daher kann die Transformation  $\Gamma_1$  für die nachfolgenden Betrachtungen ausgeschlossen werden.

Weiters können nur Transformationen, die auch die unabhängigen Variablen  $x$  und  $z$  enthalten verwendet werden um Einschränkungen für die Modenwahl zu erhalten. Damit scheiden  $\Gamma_2$  und  $\Gamma_6$  ebenfalls aus. Das ist intuitiv auch leicht verständlich, denn die Zeittranslation verschiebt nur den absoluten Startpunkt des Systems, die nachfolgende Entwicklung des Systems bleibt dadurch jedoch unverändert.

Genauso ändert die Eichungssymmetrie nur den Mittelwert der Stromfunktion, dieser ist für die Dynamik unerheblich und kann so auch im low-order model keine Rolle spielen. Wenn jetzt noch berücksichtigt wird, dass die Translationen die Randbedingungen in  $z$ -Richtung auf sich selbst abbilden müssen, bleiben nur noch zwei voneinander unabhängige Transformationen übrig, die mit dem Konvektionsrandwertproblem kompatibel sind. Diese bestehen aus Translationen in  $x$ -Richtung (d.h.  $h(t) = 1$ ,  $\varepsilon_5 = \varepsilon$ ) bzw. einer Kombination obiger diskreter Symmetrie  $e_1$  mit Spezialfällen der übrigen kontinuierlichen Symmetrien (genauer:  $\varepsilon_3 = -1$ ,  $\varepsilon_4 = 0$ ,  $h(t) = 1$  und  $\varepsilon_5 = \varepsilon$ ):

$$(t, x, z, \Psi, T) \mapsto (t, x + \varepsilon, z, -\Psi, T) \quad (4.17)$$

und

$$(t, x, z, \Psi, T) \mapsto (t, x + \varepsilon, 1 - z, -\Psi, -T) \quad (4.18)$$

## Nambu Struktur der Konvektionsgleichungen

Der konservative Teil der Gleichungen (4.1) und (4.2) schreibt sich in Nambu Form:

$$\frac{\partial \zeta}{\partial t} = - \left[ \frac{\delta \mathcal{C}}{\delta T}, \frac{\delta \mathcal{H}}{\delta \zeta} \right] - \left[ \frac{\delta \mathcal{C}}{\delta \zeta}, \frac{\delta \mathcal{H}}{\delta T} \right] = \{\zeta, \mathcal{C}, \mathcal{H}\} \quad (4.19)$$

$$\frac{\partial T}{\partial t} = - \left[ \frac{\delta \mathcal{C}}{\delta \zeta}, \frac{\delta \mathcal{H}}{\delta \zeta} \right] = \{T, \mathcal{C}, \mathcal{H}\} \quad (4.20)$$

Dabei lassen sich die Erhaltungsgrößen

$$\mathcal{H} = \int_{\Omega} \left( \frac{1}{2} (\nabla \Psi)^2 - R\sigma T z \right) df \quad (4.21)$$

und

$$\mathcal{C} = \int_{\Omega} \zeta_{(T-z)} df \quad (4.22)$$

identifizieren, wobei sich wieder  $\mathcal{H}$  auf die Gesamtenergie des Systems bezieht und  $\mathcal{C}$  wieder eine mit der Zirkulation verwandte Größe ist.

Die Nambuklammer für (4.19) und (4.20) ist für beliebige Funktionale  $\mathcal{F}_i = \mathcal{F}_i[\zeta, T]$ ,  $i = 1, 2, 3$  definiert durch

$$\{F_1, F_2, F_3\} := - \int_{\Omega} \left( \frac{\delta \mathcal{F}_1}{\delta T} \left[ \frac{\delta \mathcal{F}_2}{\delta \zeta}, \frac{\delta \mathcal{F}_3}{\delta \zeta} \right] + \frac{\delta \mathcal{F}_1}{\delta \zeta} \left( \left[ \frac{\delta \mathcal{F}_2}{\delta T}, \frac{\delta \mathcal{F}_3}{\delta \zeta} \right] + \left[ \frac{\delta \mathcal{F}_2}{\delta \zeta}, \frac{\delta \mathcal{F}_3}{\delta T} \right] \right) \right) df \quad (4.23)$$

Vergleichbar mit Kap. 3.2.3 tritt auch hier noch eine Erhaltungsgröße

$$\mathcal{S} = \int_{\Omega} g df \quad (4.24)$$

auf, diese wird jedoch nur zur Beschreibung des dissipativen Systems benötigt.

## Maximale Reduktion

Die Fourierreihenentwicklungen für die Stromfunktion und die Temperaturabweichung, die für die angenommenen Randbedingungen verwendet werden können, lauten:

$$\Psi = \sum_{n=0}^{\infty} \sum_{m=1}^{\infty} (\phi_{nm} \sin an\pi x + \varphi_{nm} \cos an\pi x) \sin m\pi z \quad (4.25)$$

und

$$T = \sum_{n=0}^{\infty} \sum_{m=1}^{\infty} (\vartheta_{nm} \sin an\pi x + \theta_{nm} \cos an\pi x) \sin m\pi z \quad (4.26)$$

Nicht alle Symmetrien aus (4.7) können realisiert werden, da der spektrale Raum diskret ist und nicht jede Verschiebungen in  $x$ -Richtung zulässt. Für die Wahl der Moden bleiben also folgende Symmetrien übrig

$$t_1 : (t, x, z, \Psi, T) \mapsto \left( t, x + \frac{1}{a}, z, \Psi, T \right) \quad (4.27)$$

und

$$t_2 : (t, x, z, \Psi, T) \mapsto (t, x, 1 - z, -\Psi, -T) \quad (4.28)$$

(4.27) und (4.28) sind nun die Symmetrien im physikalischen Raum, diese müssen nun auf den spektralen Raum übertragen werden. In den Fourierreihen kommen sie durch die additive Komponente  $+n\pi$  im  $\sin$  bzw.  $\cos$  zur Geltung.

$$\Psi = \sum_{n=0}^{\infty} \sum_{m=1}^{\infty} (\phi_{nm} \sin(an\pi x + n\pi) + \varphi_{nm} \cos(an\pi x + n\pi)) \sin m\pi z \quad (4.29)$$

und

$$T = \sum_{n=0}^{\infty} \sum_{m=1}^{\infty} (\vartheta_{nm} \sin(an\pi x + n\pi) + \theta_{nm} \cos(an\pi x + n\pi)) \sin m\pi z \quad (4.30)$$

Eine Verschiebung der Sinus- bzw. Cosinusfunktion um ein Vielfaches von  $\pi$  wirkt sich in einem Wechsel des Vorzeichens aus, die korrespondierenden Transformationen im spektralen Modell lauten also

$$t_1 : (\phi_{nm}, \varphi_{nm}, \vartheta_{nm}, \theta_{nm}) \mapsto ((-1)^n \phi_{nm}, (-1)^n \varphi_{nm}, (-1)^n \vartheta_{nm}, (-1)^n \theta_{nm}) \quad (4.31)$$

$$t_2 : (\phi_{nm}, \varphi_{nm}, \vartheta_{nm}, \theta_{nm}) \mapsto ((-1)^m \phi_{nm}, (-1)^m \varphi_{nm}, (-1)^m \vartheta_{nm}, (-1)^m \theta_{nm}) \quad (4.32)$$

Diese beiden Transformationen führen zur Einschränkung der möglichen Moden, da für alle  $n = 2k - 1$  und  $m = 2k - 1$  die Symmetrieeigenschaften verletzt werden würden.

Als diskrete Symmetriegruppe  $G$  aus den beiden Transformationen  $t_1$  und  $t_2$  resultiert

$$G = \{e, t_1, t_2, t_1 t_2\} \quad (4.33)$$

wobei mit  $e$  die Identitätstransformation gemeint ist.

Es ergeben sich die nichttrivialen Subgruppen  $S_1, S_2$  und  $S_3$ :

$$S_1 = \{e, t_1\}$$

$$S_2 = \{e, t_2\} \tag{4.34}$$

$$S_3 = \{e, t_1 t_2\}$$

Um ein System zu bekommen, in dem die drei Moden des Lorenz(1963)-Modells enthalten sind muss die Subgruppe  $S_3$  angewandt werden, bei deren Verwendung die Moden  $\phi_{11}, \varphi_{11}, \vartheta_{11}, \theta_{11}, \varphi_{02}$  und  $\theta_{02}$  übrig bleiben. Diese Koeffizienten müssen noch skaliert werden:  $\phi_{11} = bA$ ,  $\varphi_{11} = bB$ ,  $\vartheta_{11} = eD$ ,<sup>1</sup>  $\theta_{11} = eE$ ,  $\varphi_{02} = cC$ ,  $\theta_{02} = fF$ , wobei die Skalierungsfaktoren noch in folgenden Relationen zueinander stehen müssen

$$c = \frac{1}{2b}$$

$$e = \frac{a^3}{\pi^2 (1 + a^2)}$$

$$f = \frac{2a^3}{\pi^2 b^2 (1 + a^2)^2}$$

damit die Nambuklammer in der Form

$$\{\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3\} := -\nabla_{\vec{\Gamma}} \mathcal{F}_1 \cdot \nabla_{\vec{\Pi}} \mathcal{F}_2 \times \nabla_{\vec{\Pi}} \mathcal{F}_3 - \nabla_{\vec{\Pi}} \mathcal{F}_1 \cdot (\nabla_{\vec{\Gamma}} \mathcal{F}_2 \times \nabla_{\vec{\Pi}} \mathcal{F}_3 + \nabla_{\vec{\Pi}} \mathcal{F}_2 \times \nabla_{\vec{\Gamma}} \mathcal{F}_3) \tag{4.35}$$

möglich ist.

$$\vec{\Pi} = \begin{pmatrix} A \\ B \\ C \end{pmatrix}$$

$$\vec{\Gamma} = \begin{pmatrix} D \\ E \\ F \end{pmatrix}$$

sind analog zu den Vektoren des Drehimpulses und der Gravitationsrichtung.

Daraus folgt das erweiterte 6-dimensionale Lorenz(1963)-Modell:

---

<sup>1</sup> $e$  bezeichnet hier nicht mehr die Identitätstransformation sondern einen einfachen Skalierungskoeffizienten.

$$\frac{dA}{dt} = \frac{a}{2b\pi(1+a^2)} ((a^2-3)\pi^3 BC + 2eR\sigma E) \quad (4.36)$$

$$\frac{dB}{dt} = -\frac{a}{2b\pi(1+a^2)} ((a^2-3)\pi^3 AC + 2eR\sigma D) \quad (4.37)$$

$$\frac{dC}{dt} = 0 \quad (4.38)$$

$$\frac{dD}{dt} = \frac{a\pi}{2be} (e\pi CE - 2b^2 f\pi BF - 2b^2 B) \quad (4.39)$$

$$\frac{dE}{dt} = -\frac{a\pi}{2be} (e\pi CD - 2b^2 f\pi AF - 2b^2 A) \quad (4.40)$$

$$\frac{dF}{dt} = \frac{abe\pi^2}{2f} (BD - AE) \quad (4.41)$$

Die Erhaltungsgrößen dieses Systems sind:

$$\mathcal{H} = \frac{1}{4ab^2\pi} ((1+a^2)b^4\pi^3(A^2+B^2) + 2\pi^3C^2 + 4Rb^2f\sigma F) \quad (4.42)$$

und

$$\mathcal{C} = -\frac{\pi}{2ab} ((1+a^2)b^2e\pi(AD+BE) + 4f\pi CF + 4C) \quad (4.43)$$

Damit lässt sich das System in Nambu Form schreiben:

$$\frac{d\vec{x}}{dt} = \{\vec{x}, \mathcal{C}, \mathcal{H}\} \quad (4.44)$$

mit

$$\vec{x} = \begin{pmatrix} A & B & C & D & E & F \end{pmatrix}^T$$

Zusätzlich wird noch die Größe

$$\mathcal{S} = \frac{R\sigma}{12a\pi} (3e^2\pi(D^2+E^2) + 6f^2\pi F + 12fF) \quad (4.45)$$

erhalten.

Die diskrete Nambuklammer (4.35) hat nun dieselbe Struktur wie die kontinuierliche (4.1).

Das Modell erhält also die Struktur des physikalischen Problems und seine Symmetrien, es kann demnach wirklich als maximal mögliche Vereinfachung der Rayleigh-Bénard-Gleichungen angesehen werden.

# 5 Anwendung und Implementierung

Literatur: [16, 17], soweit nicht anders angegeben.

Wie schon erwähnt können reduzierte Gleichungssysteme Einblicke in physikalische Vorgänge gewähren, die beim Betrachten des vollständigen, unendlich-dimensionalen Modells aufgrund der mathematischen Komplexität verwehrt bleiben.

Besonders in früheren Zeiten, in denen die Rechenleistung der Computer noch weit geringer als die heutige war, in der sogenannten *presupercomputer era*, war eine massive Einschränkung der Modellgleichungen notwendig. Aber auch heute noch kann auf diese Weise vieles gelernt und getestet werden, Anwendungen sind außerdem im Bereich der Datenassimilation, stochastischer Parametrisierung und Ensemblevorhersage zu finden. [7]

Gerade in der Meteorologie sind solche Ensemblevorhersagen für eine realistische Bewertung der Prognosen von höchster Relevanz. Der Grundgedanke dahinter ist die Problematik der Bestimmung des Anfangszustands der relevanten Parameter für die Modellintegration. Wenn auch in den letzten Jahrzehnten große Fortschritte in der globalen Datenbeschaffung durch den Ausbau des Fernerkundungsnetzes verzeichnet werden konnten, sind die Messwerte trotz aufwändiger Assimilationsverfahren nie als 100% korrekt anzusehen. Die Ensemblevorhersage berücksichtigt diese Unsicherheit durch Generierung fiktiver Störungen, mit deren Integration gezeigt werden soll, welche Auswirkungen gewisse Abweichungen im Anfangszustand auf den Endzustand haben können.

Mit diesen Ergebnissen können Wahrscheinlichkeitsangaben für das Eintreten bestimmter Zustände gemacht werden und in heiklen Situationen auch Entwicklungen erfasst werden, die der Hauptlauf, also die ungestörte Prognose, „übersehen“ würde.

Mit den Gleichungen des erweiterten Lorenz(1963)-Modells wurde ein solches Ensemblevorhersagesystem für diese Arbeit implementiert. Hauptaugenmerk liegt auch hier wieder auf der zentralen Bedeutung der Erhaltungsgrößen.

Diese sind nicht nur bei der Herleitung von reduzierten Gleichungssystemen von großem Nutzen, sondern spielen auch in der Modellanwendung eine bedeutende Rolle.

Abgesehen davon, dass die Erhaltung von zum Beispiel der Gesamtenergie eines konservativen Systems für eine längerfristige Vorwärtsintegration Voraussetzung ist, können

sie auch zur Generierung von Störungen für die Ensemblevorhersage verwendet werden. Dieses Konzept wird im folgenden noch genauer erläutert.

Vorgreifend sei jedoch schon gesagt, dass die Generierung der Störungen unter Verwendung mehrerer Erhaltungsgrößen eine bessere Berücksichtigung der Systemgeometrie bewerkstelligen soll.

Störungen, die im Zusammenhang mit der Hamilton-Funktion berechnet werden, garantieren, dass bei deren Wahl die Erhaltung der Gesamtenergie nicht vernachlässigt wird. Kombinationen mehrerer Erhaltungsgrößen sollen diesen Vorteil ausweiten, besonders eine Verknüpfung von Energie- und Vorticityerhaltung käme den Gesetzen der realen Atmosphäre sehr nahe.

Gesucht wird also speziell nach den Störungen, die unter Erhaltung der genannten Größen die weitesten Abweichungen im integrierten Endzustand liefern, deren Trajektorie also den Attraktor nicht verlassen. Andere Variationen im Anfangszustand können zwar unter Umständen größere Schwankungsbreiten im Endzustand liefern, sind aber aufgrund der Reduktion der möglichen Zustände auf den Attraktor nicht realisierbar und führen somit zu schlechteren Ensemblevorhersagen.

### 5.1 Programmaufbau im Überblick

Programmiert wurde das EPS in Matlab 7.10.0 (R2010a) und kann ab Seite 189 vollständig nachgelesen werden.

Das Programm besteht aus zwei Teilfunktionen:

1. `rbc_6dim_autom.m`: wird zu Aufrufen von `rbc-6dim.m` verwendet und stellt sämtliche Modellparameter zur Verfügung
2. `rbc_6dim.m`: berechnet die Systementwicklung und führt die Verifikation durch

Grafisch dargestellt lässt sich der Programmablauf folgendermaßen zeigen (siehe Abb. 5.1):

## 5 Anwendung und Implementierung

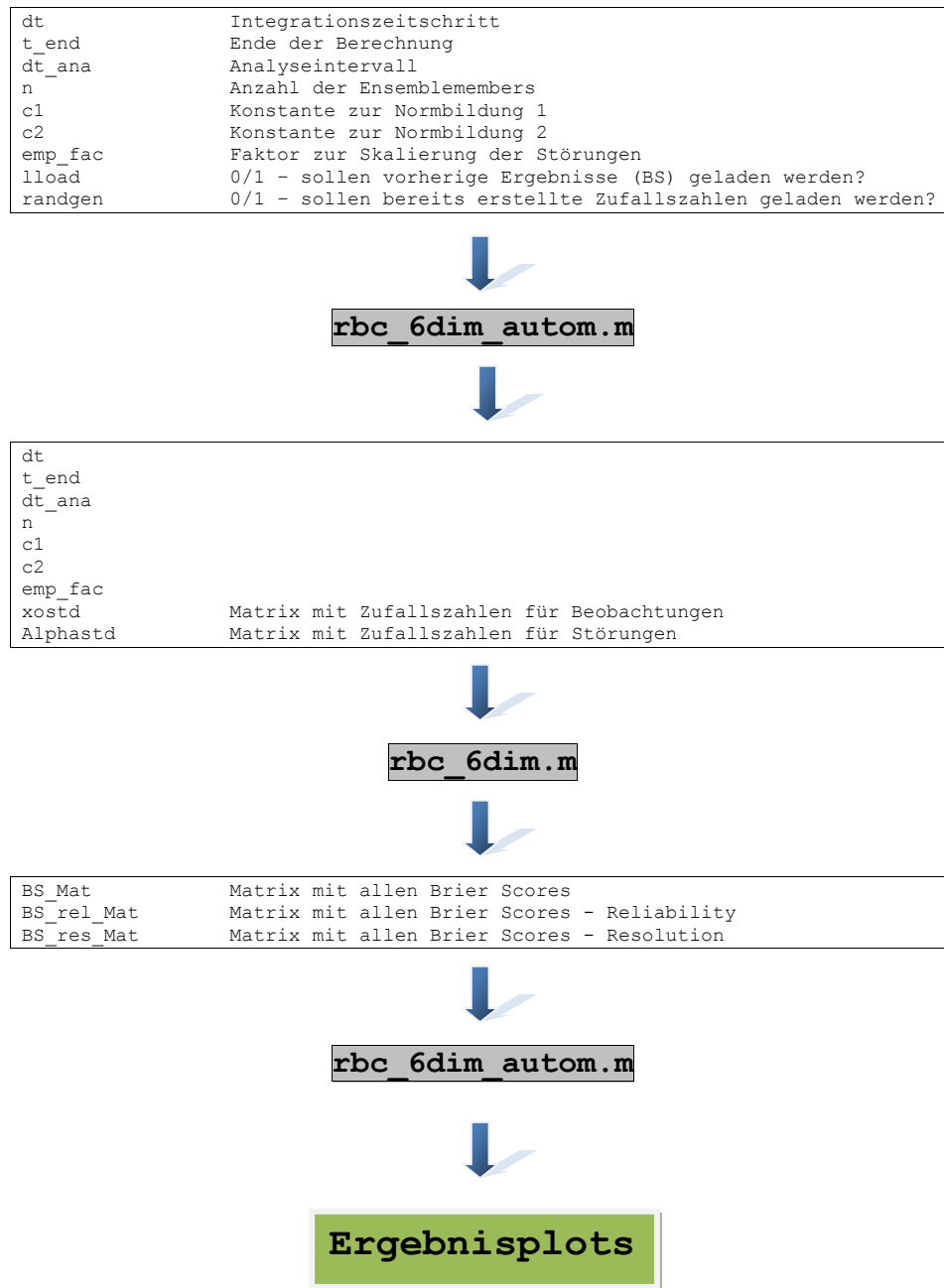


Abbildung 5.1: Darstellung der Programmstruktur

Verwendet wird ausschließlich der konservative Kern des Modells um die Auswirkungen der Erhaltungsgrößen deutlicher darstellen zu können. Die Systemgleichungen für das



vollständige Modell können in [7] nachgelesen werden.

**rbc\_6dim\_autom.m** ruft `rbc_6dim.m` beliebig oft mit frei zu wählenden Parameterkonfigurationen auf. Konstanten für die neuen Normen können variiert werden um die optimale Linearkombination aus Energie-, Helizitäts- und Temperatur-Casimir zu ermitteln.

Außerdem werden die Zufallszahlen zur Generierung von Beobachtungen und Störungen nur einmal berechnet, für jeden Programmdurchlauf werden dieselben Zufallsmatrizen übergeben um die Ergebnisse exakt vergleichbar zu halten.

**rbc\_6dim.m** gliedert sich in folgende Teile:

```
%      Modellparameter
%      Variablendefinition
%      Berechnung
%      Verifikation
%      Plots
%      Functions
```

Zuerst wird das Modell über den gesamten Betrachtungszeitraum integriert und durch kleine, zufällige Abweichungen werden Beobachtungen entlang dieser ungestörten Zeitreihe generiert.

Dann wird neuerlich integriert, diesmal werden in bestimmten Zeitintervallen Analysen durchgeführt. Die vorhergesagten Werte werden mit den zuvor generierten Beobachtungen verglichen, je nach Unsicherheit gewichtet und durch Anwendung des Kalmanfilters wird der Zustand aktualisiert.

Mittels Singulärwertzerlegung werden Störungen zur Ensemblevorhersage generiert und deren Größe durch Zufallszahlen bestimmt. Diese Störungen werden ebenfalls zeitlich entwickelt.

Zur Verifikation und Bewertung des EPS werden die Störungsvorhersagen mit den Analysen verglichen und daraus diverse Verifikationsmaße berechnet bzw. Diagramme erstellt.

## 5.2 Der Programmcode und theoretische Hintergründe

### 5.2.1 rbc\_6dim\_autom.m

Zu Beginn des Codes werden die Übergabeparameter für `rbc_6dim.m` definiert:

```
dt=0.001;
t_end=20;
dt_ana=0.1;
n=30;
c1=-10:0.5:10;
c2=-10:0.5:10;
emp_fac=1;
```

Diese haben folgende Bedeutung:

Parameter	Bedeutung
dt	Zeitschritte der nichtlinearen manuellen Integration
t_end	zeitliches Ende der Prognosen
dt_ana	Zeitintervalle der Analysen
n	Anzahl der Ensemblemembers
c1	Koeffizient der Linearkombination der Norm: für Helizität
c2	Koeffizient für die Linearkombination der Norm: für Temperatur
emp_fac	empirischer Faktor zur Feinjustierung der Störungsgrößen

Tabelle 5.1: Bedeutung der Modellparameter

Die ersten vier Parameter werden für alle Programmläufe konstant gehalten, ihre Werte können dem obigen Code entnommen werden. c1 und c2 werden auf der Suche nach der optimale Norm variiert, emp\_fac soll in etwa so gewählt werden, dass Ensemblespread und rms-Fehler bei Betrachtung der ganzen Zeitreihe im Mittel größenmäßig übereinstimmen.

Als nächstes wird überprüft ob Zufallsmatrizen generiert oder bereits vorhandene verwendet werden sollen (gesteuert durch den Parameter randgen) :

```
if (randgen)
    t_end_std=30;
    dt_ana_std=0.1;
    n_std=50;

    xostd=randn(6,t_end_std/dt_ana_std);
    Alphastd=random('norm',0,1,2,n_std);
    Alphastd(Alphastd>3)=3;
    Alphastd(Alphastd<-3)=-3;

    save('randarray')
```

```

else
    load('randarray')
end

```

Die Parameter `t_end_std`, `dt_ana_std` und `n_std` sind mit maximal verwendeten Werten für die durchgeführten Berechnungen definiert. Wird jedoch nur bis zum Beispiel `t_end=10` in `rbc_6dim.m` integriert, so stören die überflüssigen Zufallszahlen nicht, da die Beobachtungen und Störungen in Schleifen generiert werden.

Für die durchgeführten Berechnungen wurde nur ein einziges `randarray` generiert und jeweils mit

```
load('randarray')
```

geladen.

Es wird das Programm `rbc_6dim.m` für alle gewünschten Kombinationen von `c1` und `c2` aufgerufen und die Ergebnisse werden in txt-Dateien abgespeichert:

```

for j=1:length(c1)
    for k=1:length(c2)
        l=l+1;
        sprintf('Lauf Nr. %d',l)
        [ BS_Mat, BS_rel_Mat, BS_res_Mat ]=...
            dipl_aktuell(dt,t_end,dt_ana,n,...
                c1(j),c2(k),emp_fac,xostd,Alphastd);
        movefile('out.txt',...
            sprintf('ergebnisse/out_c1=%g,c2=%g.txt',c1(j),c2(k)))
        movefile('out2.txt',...
            sprintf('ergebnisse/out_c1=%g,c2=%g - good.txt',c1(j),c2(k)))
        movefile('out.pdf',...
            sprintf('ergebnisse/out_c1=%g,c2=%g.pdf',c1(j),c2(k)))
        ...
    end
end

```

Diverse Brier Scores werden in 3d-arrays abgespeichert:

```
BS_norm1(:,j,k)=BS_Mat(1,:);  
BS_norm2(:,j,k)=BS_Mat(2,:);  
...
```

Der erste Index entspricht den sechs Komponenten, der zweite und dritte speichert die verschiedenen  $c_1$ ,  $c_2$ .

Geplottet werden:

- Contourplot  $BS, c_1, c_2$  - jede Komponente separat
- Contourplot  $BS, c_1, c_2$  - Norm über alle Komponenten
- Plot  $BS, c_1$ ; norm1 vs. norm2 - jede Komponente separat
- Plot  $BS_{rel}, c_1$ ; norm1 vs. norm2 - jede Komponente separat
- Plot  $BS_{res}, c_1$ ; norm1 vs. norm2 - jede Komponente separat
- Plot  $BS, c_1$ ; norm1 vs. norm2 - Norm über alle Komponenten
- Plot  $BS_{rel}, c_1$ ; norm1 vs. norm2 - Norm über alle Komponenten
- Plot  $BS_{res}, c_1$ ; norm1 vs. norm2 - Norm über alle Komponenten

### 5.2.2 rbc\_6dim.m

#### 5.2.2.1 Modellparameter

Es werden diverse Parameter festgelegt:

Der Anfangszustand:

```
Xn=[1 1 1 1 1 1]';
```

Soll das aperiodische Verhalten der turbulenten Konvektion gezeigt werden, so ist die Wahl des Anfangszustandes nicht von großer Bedeutung. Außer der in Kap. 2.2 beschriebenen, ausgezeichneten Punkte gibt es keine Zustände, die das weitere Verhalten der Trajektorie maßgeblich einschränken. Wird ein Zustand gewählt, der im Phasenraum nicht auf dem Attraktor liegt, so bewegt sich die Trajektorie trotzdem auf schnellstem Wege zu diesem und nimmt dann ihren Verlauf laut Systemgleichungen.

Verschiedene Vorhersageschritte:

```
fc_steps=dt_ana;  
%fc_steps=dt_ana.*[1 5];
```

Als Vorhersageschritte können entweder nur das Analyseintervall oder alle ganzzahligen Vielfache davon gewählt werden, der gesamte Code wird dann für jeden Vorhersageschritt separat durchgerechnet.

Auch für zum Beispiel einen Vorhersageschritt über drei Analyseschritte wird für jeden einzelnen Analysezeitpunkt eine Prognose berechnet.

Beobachtungsskalierung:

```
p=0.03;
```

Hier wird festgelegt, wie weit die Beobachtungen von der freien Vorhersage abweichen sollen. Es wird für jede Komponente die Schwankungsbreite über den gesamten Integrationszeitraum berechnet,  $p = 0.03$  bedeutet 3% dieser Breite wird für die Beobachtung als Standardabweichung (Normalverteilung mit Mittelwert=0) verwendet.

```
global a b c e f r sig  
a=1;  
b=1;  
c=1/(2*b);  
e=a^3/(a^2*(1+a^2));  
f=2*a^3/(a^2*b^2*(1+a^2)^2);  
sig=10;  
r=28;
```

Die Modellparameter werden nach Vorgaben aus Kap. 4.1 gewählt, die relative Rayleigh-Zahl und die Prandtl-Zahl so angeben, dass sich laut Kap. 2.2 die gewünschten Prozesse einstellen können.

### 5.2.2.2 Variablendefinition

Es werden alle Variablen, die später in Schleifen berechnet werden vordefiniert um die Rechenzeit zu minimieren. Dazu werden Vektoren, Matrizen und Tensoren höheren Rangs mit Null-Einträgen verwendet damit der benötigte Speicherplatz für die arrays nicht bei jedem Schleifendurchlauf erweitert werden muss.

### 5.2.2.3 Berechnung

Die Reihenfolge, in der die einzelnen Codeabschnitte im folgenden behandelt werden, ist nicht chronologisch nach Programmablauf sondern theorieaufbauend gewählt.

In Matrixschreibweise bekommt das System die Form

$$\frac{d}{dt} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \underbrace{A(\vec{x})}_{=F(\vec{x})} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \quad (5.1)$$

$A(\vec{x})$  ist die  $6 \times 6$ -Matrix mit den Koeffizienten der Evolutionsgleichungen für  $F(\vec{x})$ , das *Forcing* des Systems.

Für (4.36) bis (4.41) lautet die Koeffizientenmatrix

$$\begin{pmatrix} 0 & \frac{a}{2b(1+a^2)}(a^2-3)\pi C & 0 & 0 & \frac{a}{b\pi(1-a^2)}eR\sigma & 0 \\ -\frac{a}{2b(1+a^2)}(a^2-3)\pi C & 0 & 0 & -\frac{a}{b\pi(1-a^2)}eR\sigma & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{ab\pi}{e} & \frac{a\pi 2b^2}{e}E & 0 & 0 & \frac{a\pi^2}{e}fbB \\ \frac{ab\pi}{e} & 0 & -\frac{a\pi^2}{2b}D & 0 & 0 & \frac{a\pi^2}{e}bfA \\ -\frac{abe\pi^2}{2f}E & \frac{abe\pi^2}{2f}D & 0 & 0 & 0 & 0 \end{pmatrix}$$

Das Forcing ist am Programmende im gleichnamigen Unterkapitel als function zu finden, auf die zur numerischen Integration zugegriffen wird.

### Numerische Integration der freien Vorhersage

Für die freie Vorhersage wird die ode45-function verwendet, die eine Mischung eines Runge-Kutta Verfahrens 4. und 5. Ordnung ist.

```
oo=odeset('RelTol',1e-7);
sol=ode45(@forcing,[0 t_end],Xn(:,1),oo);
odeget(sol,'RelTol');
Xnr=deval(sol,0:dt:t_end);
```

Die Integrationszeitschritte werden dabei nicht fix vorgegeben sondern von der function selbst gewählt um die angegebene Rechengenauigkeit einzuhalten.

## Numerische Integration und Berechnung des Tangentenpropagators mittels Heun-Schema

Das Heun-Schema gehört zu den sogenannten *singlestep-multistage methods*, für die Berechnung des folgenden Zustandes wird zwar nur eine Zeitebene verwendet, dafür aber in mehreren Schritten gerechnet (auch Prediktor-Korrektor-Schema genannt).

$$x_{(t^*)} = x_{(t_0)} + \alpha \Delta t F(x_{(t_0)}) \quad (5.2)$$

$$x_{(t)} = x_{(t_0)} + \beta \Delta t F(x_{(t^*)}) + (1 - \beta) \Delta t F(x_{(t_0)}) \quad (5.3)$$

$\alpha$  und  $\beta$  steuern die Zeitaufteilung zwischen dem Zwischenschritt  $x_{(t^*)}$  und dem Vorhersageschritt  $x_{(t)}$ . Wenn  $\alpha\beta = \frac{1}{2}$ , dann wird das Schema auch Runge-Kutta Verfahren genannt, durch Einfügen mehrerer Zwischenschritte ergeben sich Runge-Kutta Verfahren höherer Ordnung mit höherer Genauigkeit.

Für die tatsächliche Vorhersage wird ein Runge-Kutta-Verfahren mit einem Zwischenschritt manuell programmiert:

```
for i=1:(t_end/dt);
    Xn_star=Xn(:,i)+dt.*forcing(0,Xn(:,i));
    Xn(:,i+1)=0.5.*Xn(:,i)+0.5*(Xn_star+dt*forcing(0,Xn_star));
    ...
end
```

Die dabei verwendete forcing function:

```
function forcing=forcing(~,X)

varf_3=a*pi*(b*f*pi*X(6)+b)/e;

forcing=[0 varf_1*X(3) 0 0 varf_2 0; ...
        -varf_1*X(3) 0 0 -varf_2 0 0; ...
        0 0 0 0 0 0; ...
        0 -varf_3 varf_4*X(5) 0 0 0; ...
        varf_3 0 -varf_4*X(4) 0 0 0; ...
        -varf_5*X(5) varf_5*X(4) 0 0 0 0]*X;
```

end

Gleichzeitig wird der lineare Tangentenpropagator  $M$  berechnet - eine Matrix, die angibt, wie vom vorigen Zeitschritt linear zum aktuellen zu kommen ist. Sie kann unter der Annahme oder der Näherung einer linearen Entwicklung auch für die Vorhersage weiterer Zeitschritte in der Zukunft verwendet werden, diese lineare Approximation gilt jedoch nur für sehr kurze Integrationszeiten und kleine Störungen.

Der Propagator wird ebenfalls mit einem Heun-Schema berechnet,

$$\begin{aligned}
 x'_{(t^*)} &= x'_{(t_0)} + J(x_{(t_0)}) \Delta t x'_{(t_0)} = (E + J(x_{(t_0)}) \Delta t) x'_{(t_0)} \\
 x'_{(t)} &= \frac{1}{2} (x'_{(t_0)} + (E + J(x_{(t_0)}) \Delta t) x'_{(t^*)}) \\
 &= \underbrace{\frac{1}{2} (E + (E + J(x_{(t^*)}) \Delta t) (E + J(x_{(t_0)}) \Delta t))}_{M(t_0, t)} x'_{(t_0)} \quad (5.4)
 \end{aligned}$$

und ist nur vom Grundzustand abhängig. Dabei ist  $J(x)$  die Jacobimatrix, die in Kürze genauer beschrieben wird, auch die function wird später vorkommen.

```
M(:, :)=0.5*(eye(num_dim)+(eye(num_dim)+dt*jacobi(Xn_star))...
    *(eye(num_dim)+dt*jacobi(Xn(:, i))));
```

Da die Integrationszeitschritte sehr viel kleiner als die Analysezeitschritte zu wählen sind, entsteht die Matrix, die von vorherigen zur aktuellen Analyse führt durch mehrmalige Multiplikation der Matrix  $M$  und wird als Resolvente  $A$  bezeichnet. Vor jedem neuen Analyseschritt muss die Resolvente auf die Einheitsmatrix zurückgesetzt werden.

```
A=M*A;
```

## Beobachtungen

Im `rbc_6dim_autom.m` wird eine Matrix mit normalverteilten Zufallszahlen generiert ( $\mu = 0$  und  $\sigma = 1$ ), diese wird mit dem Prozentsatz  $p$  der Gesamtschwankungsbreite der jeweiligen Komponente skaliert. Durch Addition dieser Matrix mit der Matrix der frei entwickelten Zeitreihe entstehen Beobachtungen, dies geschieht jedoch erst in der Analyseschleife.



```

%xo=randn(num_dim,t_end/dt_ana);
xo=xostd;
for m=1:num_dim
    I(m)=range(Xnr(m,:));
    I_obs(m)=p.*I(m);
    xo(m,:)=xo(m,:)*I_obs(m);
end

...
x_obs(:,k)=Xnr(:,i+1)+xo(:,k);

```

## Analysen und Kalmanfilter

Die Analysen finden in der Schleife

```

if mod((i),(dt_ana/dt))==0
k=k+1

```

statt und laufen mit dem Index  $k$ .

Das Kalmanfilter wird verwendet um neue Informationen, in diesem Fall aktuelle Beobachtungen, in die Systementwicklung einzubinden. Es stammt ursprünglich aus der linearen Schätztheorie, hier wird eine erweiterte Variante, das *Extended Kalmanfilter* benutzt. Ausschlaggebend ist dabei eine optimale Gewichtung zwischen Vorhersage und Zusatzinformation. [22]

Dazu werden Fehlerkovarianzmatrizen aufgestellt:

Symbol	Name	Bedeutung
$P$	Backgroundfehlerkovarianzmatrix	Unsicherheit der Vorhersage
$R$	Beobachtungsfehlerkovarianzmatrix	Unsicherheit der Beobachtungen
$Q$	Modellfehlerkovarianzmatrix	Unsicherheit des Modells
$K$	Kalmanfilter	gewichtet Innovation
$E$	Einheitsmatrix	

Tabelle 5.2: Fehlerkovarianzmatrizen

Je größer eine Fehlerkovarianz, umso weniger wird die entsprechende Größe gewichtet. Ist zum Beispiel bekannt, dass die Beobachtungen sehr ungenau sind, so werden diese weniger gewichtet und die Analyse, also der “wahre Wert” liegt eher bei der Vorhersage.

Die Fehlerkovarianzmatrizen werden im Code definiert:

```
R=diag(I_obs.^2,0);
Q=diag(I_obs.^2.*1.2,0);
P=diag(I_obs.^2,0);
```

$Q$  wird so gewählt, dass die Unsicherheit des Modells etwas größer ist, als die der Beobachtungen. Für  $P$  wird zwar auch ein Startwert definiert, diese Größe dient jedoch nur als Ausgangswert, die Matrix wird dann ohnehin im Kalmanfilter bei Analyse berechnet.

Die Vorhersage des Zustands entstammt den nichtlinearen Vorhersagegleichungen:

```
x(:,k)=Xn(:,i+1);
```

Die Backgroundfehlerkovarianz wird durch das tangentiallineare Modell, also den linearen Tangentenpropagator  $M$ , der in der Resolvente  $A$  impliziert ist, berechnet:

```
P=A'*P*A+Q;
```

bzw. mit Josephs Erweiterung zur Stabilisierung [20]:

```
P=(E-A)'*P*(E-A)+A'*R*A+Q;
```

Das Kalmanfilter ergibt sich aus

```
K=P/(P+R)
```

und wird für den Update-Schritt

```
x(:,k)=x(:,k)+K*(x_obs(:,k)-x(:,k));
```

gebraucht. Die Größe

```
(x_obs(:,k)-x(:,k));
```

wird *Innovation* genannt, das Kalmanfilter  $K$  gewichtet nun diese neue Information. Da die Backgroundfehlerkovarianzmatrix  $P$  zu jedem Analyseschritt neu berechnet wird, kann das Kalmanfilter berücksichtigen, wie sensibel die Vorhersage gerade ist und bei hoher Fehleranfälligkeit der Beobachtung mehr Relevanz zuschreiben.

Zuletzt wird noch die Fehlerkovarianzmatrix aktualisiert:

```
P=P-K*P;
```

Die gewonnene Analyse  $x$  wird für die weitere Integration verwendet, der Zustand wird nichtlinear weiterentwickelt und erst beim nächsten Analysezeitpunkt wieder durch Informationen neuer Beobachtungen aktualisiert.

### Ensemblevorhersage - Konzept

Um die Unsicherheit einer Zustandsprognose abschätzen zu können kann eine Ensemblevorhersage durchgeführt werden. Dabei wird der Ausgangspunkt der Integration leicht gestört und am Ende der Integration die Auswirkung dieser Störung analysiert. Bewirken minimale Änderungen im Anfangszustand signifikante Abweichungen im Endzustand, so ist die Vorhersage nur mit Vorsicht zu genießen, es bleibt immer zu bedenken, dass es nie einen 100% richtigen physikalischen Messwert für eine Ausgangssituation geben kann.

In dem als deterministisches Chaos identifizierten Lorenz(1963)-Modell divergieren Störungen grundsätzlich, es stellt sich nur die Frage wie sehr. In Abb. 2.8 wurde bereits demonstriert, dass auf dem Attraktor sensiblere und weniger sensible Bereiche bezüglich Störungen existieren.

In der kommerziellen Zustandsvorhersage der Atmosphäre steht nicht so sehr die Betrachtung der Vorhersageunsicherheit, sondern eher die daraus resultierende Frage nach dem Bereich des Möglichen im Vordergrund. Hier versucht die Ensemblevorhersage dem Konsumenten die Information zu liefern, womit er rechnen muss / worauf er vorbereitet sein sollte.

Dieses Konzept klingt zunächst sehr einfach, bei der Realisierung solcher EPS stellt sich jedoch relativ bald die Frage nach den zu wählenden Störungen um wirklich den “gesamten” Bereich des Möglichen abzudecken.

Ein mächtiges Werkzeug für diese Fragestellung liefert die Mathematik mit der Singulärwertzerlegung. Steht die Matrix  $A$ , die den Weg von einem Zustand zum darauf folgenden linear approximiert, zur Verfügung, so können mittels der Zerlegung  $A = U\Sigma V^T$  die singulären Vektoren ermittelt werden, die von der Matrix maximal gestreckt werden. Das sind die Störungen, die zur maximalen Lösungsbandbreite führen, in *Matlab* existiert für deren Berechnung die Funktion  $svd(A)$ .

Doch was ist die maximale Lösungsbandbreite? Was ist der größte Fehler?

Prognostiziert man den Zustand der Atmosphäre, so ist ein Fehler einer einzigen Größe, zum Beispiel der Temperatur zwar tragisch, für das Modell selbst und dessen weitere Integration wäre aber ein großer Fehler, beispielsweise die Gesamtenergie betreffend, weit schlimmer. Natürlich kann in Modellen, die die reale Erdatmosphäre beschreiben sollen, nicht auf rein konservative Systeme zurückgegriffen werden, die die Gesamtenergie komplett erhalten.

Derzeit werden die EPS am ECMWF unter Berücksichtigung einer Energiegröße zur Störungsgenerierung durchgeführt. [12] Da die Grundgleichungen zur Beschreibung der Atmosphäre auf ein nichtkanonisches Hamiltonsches System mit zusätzlichen Casimiren

zurückgeführt werden können, wäre es theoretisch möglich, dass eine Erweiterung der *Energienorm* auf eine *Energie-Casimir-Norm* zur Optimierung der EPS beitragen kann.

Den Vorteil niedrig-dimensionaler Modelle ausnutzend, soll diese Theorie nun anhand des erweiterten Lorenz(1963)-Modells getestet werden, das durch seine Nambu-Form die Erhaltung der Energie sowie der beiden Casimire gewährleistet. Berechnungen unter der Berücksichtigung einer reinen Energienorm, in diesem Fall der Hamilton-Funktion  $\mathcal{H}$  werden mit kombinierten Größen  $\mathcal{H} + c_1\mathcal{C}$  und  $\mathcal{H} + c_1\mathcal{C} + c_2\mathcal{S}$  verglichen. An dieser Stelle wird auch der Vorteil des erweiterten 6-dimensionalen Modells gegenüber dem ursprünglichen Lorenz-Gleichungen von 1963, die ausschließlich energiebezogene Erhaltungsgrößen bieten können, deutlich. Zwar wird grundsätzlich jede zusätzliche erhaltene Größe neben der Hamilton-Funktion als Casimir bezeichnet, allerdings soll im folgenden speziell der Nutzen von Vorticity-Casimiren getestet werden, basierend auf Erkenntnissen der *Energie-Wirbel-Theorie*.

In Anlehnung an die geometrische Interpretation der Erhaltungsgrößen des Lorenz-Modells in Abb. 3.1 kann die Verwendung der Erhaltungsgrößen zur Störungsberechnung ebenfalls geometrisch erklärt werden.

Werden die Störungen durch die Singulärwertzerlegung der Resolvente alleine berechnet so liefert dies die Vektoren, deren Eulernorm bei linearer Entwicklung am meisten anwächst. Das können aber auch singuläre Vektoren in Richtungen sein, die vom System aufgrund der Zwangsbedingungen der Erhaltungsgrößen unter Umständen gar nicht realisiert werden können. Die entwickelten Störungen lägen also nicht mehr am Attraktor. Wenn aber für die Störungsberechnung beispielsweise die Hamilton-Funktion verwendet wird, kann diese Zusatzbedingung, zumindest die der Energieerhaltung, berücksichtigt werden.

Ein solcher Fall ist anhand des Lorenz-(1963)-Attraktors in Abb. 5.2<sup>1</sup> dargestellt.

---

<sup>1</sup>Quelle: Code auf Seite 175

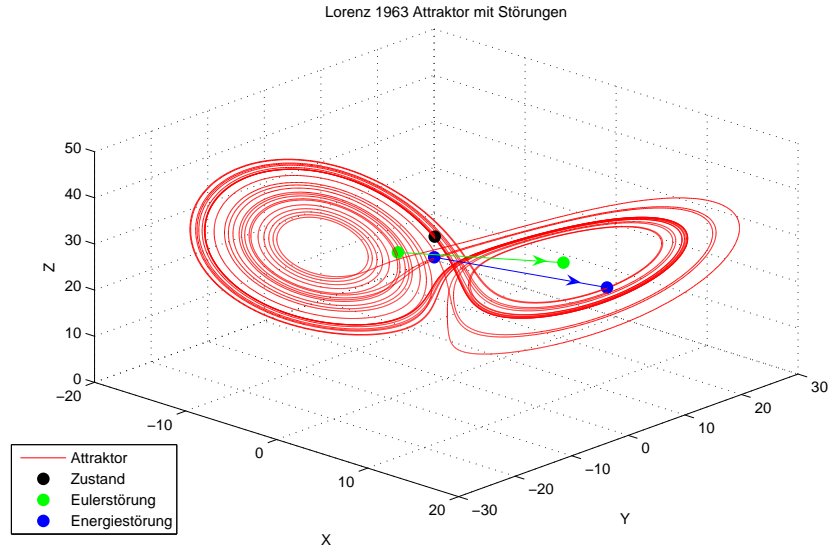


Abbildung 5.2: Verschiedene Störungsnormen am Lorenz(1963)-Attraktor

Grün eingezeichnet ist die Störung, die durch die reine Eulernorm gebildet wurde und blau die Störung unter Berücksichtigung einer Erhaltungsgröße.

### Ensemblevorhersage - Störungsgenerierung

Die Singulärwertzerlegung erfolgt also nicht rein bezüglich der Resolvente  $A$  sondern diese wird durch Erhaltungsgrößen modifiziert:

Die Normberechnung lässt sich im Allgemeinen schreiben als

$$Norm = \sqrt{\vec{x}^T W^T W \vec{x}} \quad (5.5)$$

wobei  $\vec{X}$  der Zustandsvektor und  $W$  eine Matrix zur Gewichtung der einzelnen Komponenten ist.

Für die Euklidische Norm gilt

$$W = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

was in 3 Dimensionen zu dem bekannten Ergebnis

$$E - Norm = \sqrt{\begin{pmatrix} x & y & z \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}} = \sqrt{x^2 + y^2 + z^2} \quad (5.6)$$

führt.

Die Berechnung des Zustandes zum Zeitpunkt  $t$  ausgehend vom Zeitpunkt  $t_0$  erfolgt mit der Resolvente  $A$ , die ja die aktuelle Systementwicklung linear approximiert, durch

$$\vec{x}(t) = A(t_0, t) \vec{x}(t_0) \quad (5.7)$$

Singuläre Vektoren bezüglich der Eulerschen Norm, werden durch die Singulärwertzerlegung der Resolvente berechnet:

$$U\Sigma V^T = \text{svd}(A(t_0, t)) \quad (5.8)$$

Die Erhaltungsgrößen des betrachteten Modells sind, wie schon in Kap. 4.1 beschrieben:

$$\mathcal{H} = \frac{1}{4ab^2\pi} \left( (1+a^2) b^4 \pi^3 (A^2 + B^2) + 2\pi^3 C^2 + 4Rb^2 f \sigma F \right) \quad (5.9)$$

$$\mathcal{C} = -\frac{\pi}{2ab} \left( (1+a^2) b^2 e \pi (AD + BE) + 4f \pi CF + 4C \right) \quad (5.10)$$

$$\mathcal{S} = \frac{R\sigma}{12a\pi} \left( 3e^2 \pi (D^2 + E^2) + 6f^2 \pi F + 12fF \right) \quad (5.11)$$

Nun wird nicht die Entwicklung des Zustandes selbst sondern beispielsweise die der Hamilton-Funktion

$$\mathcal{H} = \vec{x}^T W_H^T W_H \vec{x} \quad (5.12)$$

betrachtet:

$$W_H^* \vec{x}(t) = W_H^* A(t_0, t) x(t_0) \quad (5.13)$$

Gemeint ist damit natürlich eine lineare Entwicklung der Hamilton-Funktion ausgehend von der linear approximierten Entwicklung des Zustands im vorherigen Zeitschritt. Im korrekt integrierten Modell müsste ja jede zeitliche Ableitung der Hamilton-Funktion und aller Casimire Null ergeben.

Um wieder auf die vorherige geometrische Interpretation zurückzukommen, stelle man sich also einen Pfeil vor, der vom aktuellen Zustand zum gestörten Zustand zeigt. Während der Integration wandert dieser Pfeil mit dem Gesamtzustand mit, wird gedreht

und gestreckt und tritt als entwickelter Pfeil an einer anderen Stelle im Phasenraums wieder auf. Diese Entwicklung wird vor der eigentlichen Integration zur Störungsgenerierung durch die Resolvente  $A$  approximiert. Nur weil  $A$  als Matrix definitionsgemäß eine lineare Abbildung ist, ist diese Aufspaltung von Entwicklung des Grundzustands und Entwicklung der Störung möglich. Ziel ist es durch diese Approximation, also unter Verwendung von  $A$ , jene gestörten Zustände zu ermitteln, für die die Pfeile am Ende maximal gestreckt werden. Für die Vorstellung von zum Beispiel 50 Störungen, sollen die 50 Pfeile am Ende der Integration<sup>2</sup> eine maximale<sup>3</sup> Fläche *am Attraktor* anzeigen. Die Singulärwertzerlegung sucht den Startpunkt / den gestörten Zustand, der den längsten Vektor bewirkt.

Gäbe es jetzt beispielsweise eine Vorschrift, die besagt, dass alle möglichen Punkte  $\vec{x}$  nur gerade Zahlen sein können, so müsste die Matrix  $A$  nur komponentenweise mit 2 multipliziert werden und wäre nie wieder im Stande aus

$$\vec{x}(t) = A(t_0, t) \vec{x}(t_0) \quad (5.14)$$

Punkte mit ungeraden Werten zu erzielen.

Wäre die Vorschrift, dass eine bestimmte Komponente immer Null sein muss, dann bräuchte die entsprechende Zeile/Spalten der Matrix  $A$  nur ebenfalls Null gesetzt werden und sie wäre nie wieder im Stande, einen Punkt zu erzeugen, bei dem diese Komponente nicht Null ist.

Die Vorschrift für das Lorenz-System kann ähnlich aufgefasst werden: es dürfen nur Punkte erreicht werden, die am Attraktor liegen. Da der Attraktor aber eine etwas kompliziertere Geometrie aufweist, wird jede Komponente der Matrix  $A$  mit einem anderen Faktor multipliziert. Da er zudem auch räumlich nicht homogen ist, hängen diese Faktoren zusätzlich vom momentanen Aufenthaltsort ab.

Diese Vorschrift wird nun durch die Gewichtungsmatrix  $W$  realisiert und lautet im Fall von  $W_H^*$ : Nicht die Flächen konstanter Energie verlassen!

In Abb. 3.1 wurden die Erhaltungsgrößen des 3-dimensionalen Modells geplottet, sie haben Ähnlichkeit mit einer Röhre und einer Dachrinne, ihre Schnittmenge ergibt den Attraktor.

Das Problem ist jedoch, dass auch eine Entwicklung, die die Erhaltung der einen Größe garantiert, also zum Beispiel auf der Röhre bleibt, trotzdem den Attraktor verlassen kann, da sie nichts von der Dachrinne weiß.

Darum soll die Kombination aus mehreren Erhaltungsgrößen bessere Störungen liefern.

---

<sup>2</sup>der tatsächlichen, nichtlinearen Integration

<sup>3</sup>eine maximal für die verwendete Größe der Störungen mögliche Fläche

Optimal wäre demnach ein Operator zur Modifikation der Resolvente, der die exakte Geometrie des Attraktors erfasst, dieser wäre jedoch sicher nicht linear und könnte somit nicht in eine einfache Matrix verpackt werden. Nachdem dieser Arbeit kein Mathematikstudium vorausging, sind der Autorin die Möglichkeiten und Grenzen der Mathematik nicht ganz bekannt, doch auf jeden Fall ist das Nicht-Verwenden bekannter Erhaltungsgrößen eigentlich fast eine "Verschwendung von Potential". Leider ist die Realisierung dieser Verwendung nicht ganz trivial. Um für die Untersuchung dieser Theorie einen Startpunkt zu setzen, werden hier nun Linearkombinationen der konservativen Größen getestet. Es ist zu hoffen, dass es empirische Faktoren für diese linearen Kombination gibt, die der tatsächlichen Attraktorgeometrie etwas näher kommen, als jede einzelne Erhaltungsgröße für sich alleine.

Um die Störungsberechnung basierend auf der Hamilton-Funktion durchzuführen, muss die Gewichtungsmatrix  $W_H$  gefunden werden, sodass die  $H - Norm$

$$H - Norm = \sqrt{\vec{x}^T W_H^T W_H \vec{x}} \quad (5.15)$$

ist.

Die Ermittlung der optimalen Störungen funktioniert dann mit

$$U \Sigma V = svd(W_H(\vec{x}(t)) A(t_0, t)) \quad (5.16)$$

Die Gewichtungsmatrizen  $W_H^*$  für die drei genannten Erhaltungsgrößen sind 6-dimensional und haben folgende Einträge um die Entwicklungsgleichung

$$W_H^* \vec{x}(t) = W_H^* A(t_0, t) \vec{x}(t_0) \quad (5.17)$$

bzw. die analogen Gleichungen für  $\mathcal{C}$  und  $\mathcal{S}$  zu erfüllen:

Hamilton-Funktion  $\mathcal{H}$  :

$$W_{H(1,1)}^* = (1 + a^2) b^4 \pi^3 A$$

$$W_{H(2,2)}^* = (1 + a^2) b^4 \pi^3 B$$

$$W_{H(3,3)}^* = 2\pi^3 C$$

$$W_{H(6,6)}^* = 4Rb^2 f \sigma$$

Der Helizitäts-Casimir  $\mathcal{C}$  :

$$W_{C(1,1)}^* = (1 + a^2) b^2 e \pi D$$

$$W_{C(2,2)}^* = (1 + a^2) b^2 e \pi E$$



## 5 Anwendung und Implementierung

$$W_{C(3,3)}^* = 4f\pi F + 4$$

Temperatur-Casimir  $\mathcal{S}$  :

$$W_{S(4,4)}^* = 3e^2\pi D$$

$$W_{S(5,5)}^* = 3e^2\pi E$$

$$W_{S(6,6)}^* = 6f^2\pi F + 12f$$

Alle anderen Einträge der  $6 \times 6$ -Matrizen werden auf 0 gesetzt, die konstanten Faktoren vor den Klammern in (5.9) bis (5.11) können auch weggelassen werden.

Für die Normberechnung in (5.15) wird die Matrix  $W_H$  benötigt, welche sich formal anschreiben lässt als

$$W_H^2(i, j) = \frac{W_H^*(i, j)}{x(j)}, \quad (5.18)$$

es müsste also nur komponentenweise durch die jeweilige Komponente dividiert und die Wurzel gezogen werden.

Hamilton-Funktion  $\mathcal{H}$  :

$$W_{H(1,1)} = \sqrt{(1 + a^2)b^4\pi^3}$$

$$W_{H(2,2)} = \sqrt{(1 + a^2)b^4\pi^3}$$

$$W_{H(3,3)} = \sqrt{2\pi^3}$$

$$W_{H(6,6)} = \sqrt{\frac{4Rb^2f\sigma}{F}}$$

Der Helizitäts-Casimir  $\mathcal{C}$  :

$$W_{C(1,1)} = \sqrt{\frac{(1 + a^2)b^2e\pi D}{A}}$$

$$W_{C(2,2)} = \sqrt{\frac{(1 + a^2)b^2e\pi E}{B}}$$

$$W_{C(3,3)} = \sqrt{\frac{4f\pi F + 4}{C}}$$

Temperatur-Casimir  $\mathcal{S}$  :

$$W_{S(4,4)} = \sqrt{3e^2\pi}$$

$$W_{S(5,5)} = \sqrt{3e^2\pi}$$

$$W_{S(6,6)} = \sqrt{6f^2\pi + \frac{12f}{F}}$$

Hier ist aber größte Vorsicht geboten, zwar ist, vor allem  $W_H$  und  $W_S$  großteils positiv definit, da unter den Wurzeln meist positive Konstanten stehen, jedoch finden sich in allen drei Gewichtungsmatrizen auch lineare Terme der Zustandskomponenten. In  $W_H$  und  $W_S$  ist die Komponente  $F$  das Problem, in  $W_C$  verhält es sich noch komplizierter. Das bedeutet bei einer Oszillation der Komponenten um den Nullpunkt, dass einzelne Matrixeinträge negativ werden und somit durch das Wurzelziehen der rein reelle Raum verlassen wird. Da die Imaginärteile für die Systemintegration keine Rolle spielen, verlieren die Normen der Erhaltungsgrößen ihre konservative Eigenschaft und bewirken große Schwankungen. Somit verliert das gesamte Konzept seine Wirksamkeit.

In `rbc_6dim.m` wird überprüft, ob die Normen während der Systemintegration erhalten bleiben und ob Imaginärteile in den Gewichtungsmatrizen entstehen:

```
for i=1:t_end/dt+1
    H_norm_Mat_check(1,i)=(H_norm(Xnr(:,i))
    *Xnr(:,i))'*(H_norm(Xnr(:,i))*Xnr(:,i));
    C_norm_Mat_check(1,i)=(C_norm(Xnr(:,i))*Xnr(:,i))'
    *(C_norm(Xnr(:,i))*Xnr(:,i));
    S_norm_Mat_check(1,i)=(S_norm(Xnr(:,i))
    *Xnr(:,i))'*(S_norm(Xnr(:,i))*Xnr(:,i));

    H_imag_norm=norm(imag(H_norm(Xnr(:,i)))));
    C_imag_norm=norm(imag(C_norm(Xnr(:,i)))));
    S_imag_norm=norm(imag(S_norm(Xnr(:,i)))));

    if H_imag_norm~=0
        disp('H')
        disp(H_imag_norm)
        u=u+1;
    end

    if C_imag_norm~=0
        disp('C')
        disp(C_imag_norm)
        u=u+1;
    end

    if S_imag_norm~=0
        disp('S')
```

```

        disp(S_imag_norm)
        u=u+1;
    end

end

figure('Units','normalized','Position',[0 0 1 1]);

subplot(3,1,1)
plot(t,H_norm_Mat_check(1,:), 'y')
title('Zeitlicher Verlauf Hamilton-Funktion, berechnet mit Norm-Matrix')
xlabel('Zeit')
ylabel('Energienorm')

subplot(3,1,2)
plot(C_norm_Mat_check(1,:), 'y')
title('Zeitlicher Verlauf Helizitaets-Casimir, berechnet mit Norm-Matrix')
xlabel('Zeit')
ylabel('Helizitaetsnorm')

subplot(3,1,3)
plot(S_norm_Mat_check(1,:), 'y')
title('Zeitlicher Verlauf Temperatur-Casimir, berechnet mit Norm-Matrix')
xlabel('Zeit')
ylabel('Temperaturnorm')

...

disp('Anzahl der imaginaer gewordenen Normen')
disp(u)

```

In Abb. 5.3<sup>4</sup> sind die fatalen Auswirkungen der Vernachlässigung der Imaginärteile zu sehen:

---

<sup>4</sup>Quelle: Code auf Seite 178

## 5 Anwendung und Implementierung

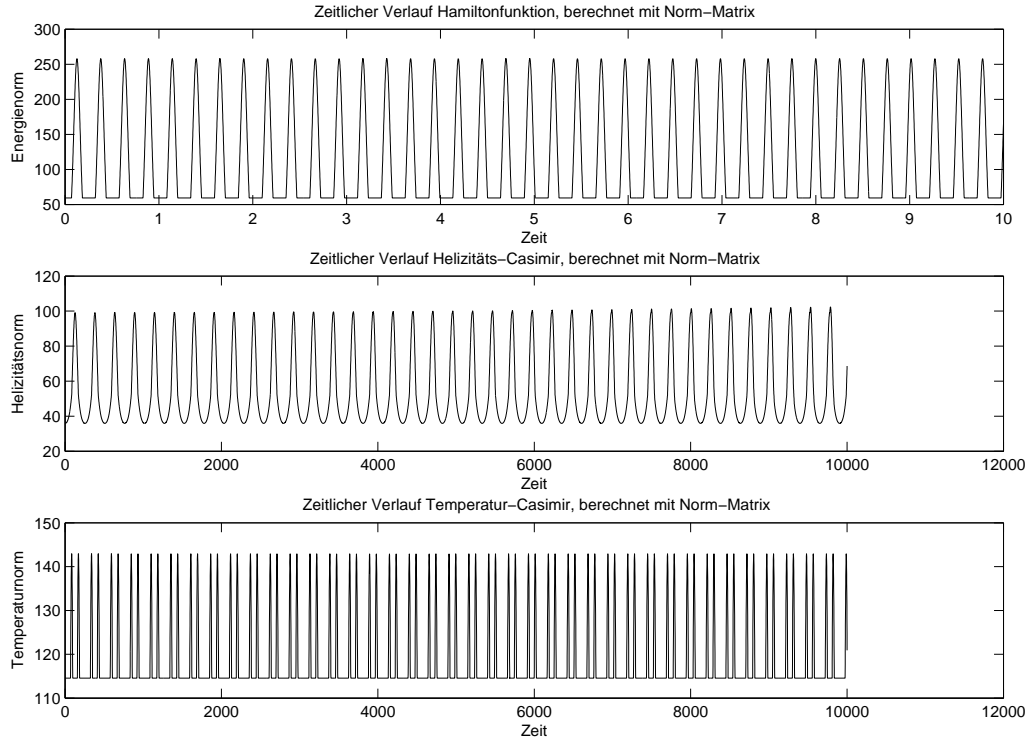


Abbildung 5.3: Zeitliche Entwicklung der Erhaltungsgrößen bei imaginären Gewichtungsmatrizen

Das Problem ist folglich, dass die Erhaltungsgrößen des untersuchten Modells eine derart komplexe Form aufweisen, dass sie erstens nicht im ganzen Vektorraum gleich sind, sondern deren Berechnungsvorschrift selbst vom aktuellen Zustand abhängt und dass zweitens diese Abhängigkeit linear ist. Die Linearität an sich ist hier nicht die Schwierigkeit, nur der Umstand, dass durch diese die Vorzeichen der Zustandskomponenten sozusagen „mit in die Norm genommen werden“. Selbiges Problem würde jede Abhängigkeit des Zustands mit ungeradem Exponenten liefern, nur gerade Potenzen der Komponenten könnten hier Abhilfe schaffen.

Die Tatsache der zustandsabhängigen Normen scheint auf den ersten Blick unheimlich, da aber der Attraktor nicht homogen ist, könnte gerade dieser Umstand für einen positiven Effekt zur Geometrieerfassung sorgen. Diese Hypothese soll hier nur in den Raum gestellt werden, deren Überprüfung bedarf wahrscheinlich aufwändigerer Untersuchungen.

Das hier auftretende Problem der negativen Diskriminanten widerlegt noch nicht die Arbeitsthese, allerdings weist es darauf hin, dass das gewählte low-order-Modell für die Untersuchung eher suboptimal ist. Nachdem die Auswahl an strukturerhaltenden reduzierten Modellen mit all den für die These benötigten Eigenschaften nicht allzu üppig

ist, soll an dieser Stelle noch nicht aufgegeben werden.

Da die Struktur der Erhaltungsgrößen feststeht, besteht die einzige Möglichkeit darin, den Attraktor des Modells derart zu manipulieren, dass seine Komponenten keine negativen Werte unter den Wurzeln der Gewichtungsmatrizen hervorrufen.

Für die Norm der Hamilton-Funktion und des Temperatur-Casimirs müsste dazu nur die Komponente  $F$  aus dem negativen Halbraum ferngehalten werden. Für die Verwendung des Helizitäts-Casimirs sollten  $A$  und  $D$ ,  $B$  und  $E$  sowie  $C$  und  $F$  stets gleiche Vorzeichen haben. Alternativ zur allerletzten Bedingung für  $C$  und  $F$  kann auch der Geometriefaktor  $f$  dazu verwendet werden, den Einfluss des Terms proportional zu  $F$  so klein zu halten, dass er durch das immer positive  $C$  (wenn im Anfangszustand derart gewählt) überkompensiert wird.  $f$  hängt von  $a$  und  $b$  ab, verändert man diese beiden Faktoren, so ändern sich allerdings automatisch auch  $c$  und  $e$ .

Ohne diese Überlegungen jetzt weiter darzulegen, wird an dieser Stelle nur darauf hingewiesen, dass derartige Wünsche die Wertebereiche und Relationen der einzelnen Komponenten betreffend, im Zusammenhang mit einem 6-komponentigen (effektiv 5-komponentigen) partiellen Differentialgleichungssystem eher anstrengend zu realisieren sind.

Es ist durch Anpassung sämtlicher Systemparameter jedoch gelungen, sowohl  $F$  positiv, als auch die Imaginärteile von  $W_C$  in Anzahl und Größe gering zu halten. Für die später durchgeführten Berechnungen zeigt eine Kontrollfunktion, dass negative Diskriminanten zu den Analysezeitpunkten sogar vollständig vermieden werden können.

Im Wesentlichen wurde dafür der Anfangszustand  $X_n = [1 \ 1 \ 4 \ 1 \ 1 \ 1]$  und der geometrische Faktor  $b = 1.102$  modifiziert, alle anderen Parameter bleiben wie in 5.2.2.1 beschrieben.

Nachdem die physikalische Sinnhaftigkeit des Modells unter Verwendung der herausgefundenen Parameterkonfiguration erhalten bleibt, kann das Konzept auf diese Weise weiter verfolgt werden. Es ist lediglich zu bemerken, dass der Vorzeichenwechsel der Komponenten etwas weniger oft stattfindet.

Die signifikante Verbesserung der Erhaltungseigenschaften wird in Abb. 5.4<sup>5</sup> deutlich, man beachte dabei die Skalierung der  $y$ -Achse:

---

<sup>5</sup>Quelle: Code auf Seite 178

## 5 Anwendung und Implementierung

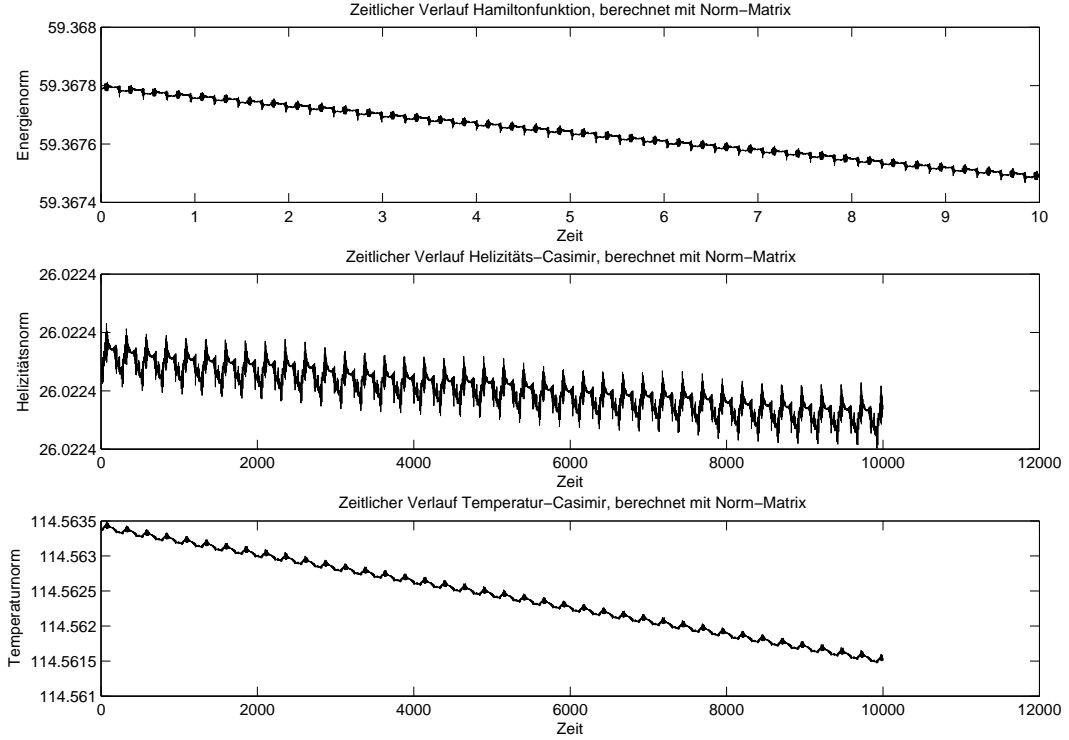


Abbildung 5.4: Zeitliche Entwicklung der Erhaltungsgrößen bei reellen Gewichtungsmatrizen

Wie oben bereits erwähnt, soll nun nicht nur eine Erhaltungsgröße verwendet werden, sondern eine Linearkombination aus mehreren.

In Anlehnung auf die reale Atmosphäre, in der die Erhaltung von Gesamtenergie und Ertelscher potentiellen Vorticity von zentraler Bedeutung sind, soll eine neue Norm der Form

$$H - Norm + c_1 \cdot C - Norm \quad (5.19)$$

$$c_1 = konst$$

gebildet werden. Sie stellt die zuvor erwähnte *Energie-Casimir-Norm* mit Vorticity-Beitrag dar.

Weiters soll eine Norm getestet werden, die alle drei Erhaltungsgrößen berücksichtigt:

$$H - Norm + c_1 \cdot C - Norm + c_2 \cdot S - Norm \quad (5.20)$$

## 5 Anwendung und Implementierung

$$c_1 = konst, \quad c_2 = konst$$

Für die Störungen kann entweder nur der erste oder eine Linearkombination aus den ersten beiden singulären Vektoren genommen werden, letzteres macht Sinn, wenn die ersten beiden Singulärwerte ähnlich groß sind.

```
[~,~,V1]=svd(norm1(Xn(:,i))*A);
V1(3,:)=0;
[~,~,V2]=svd(norm2(Xn(:,i))*A);
V2(3,:)=0;
[~,~,V3]=svd(norm3(Xn(:,i))*A);
V3(3,:)=0;

sigma=sqrt(trace_P_after_update(:,k));
Alpha=Alphastd*sigma;
%   Alpha=random('norm',0,sigma,2,n);
%   Alpha(Alpha>3*sigma)=3*sigma;
%   Alpha(Alpha<-3*sigma)=-3*sigma;
%   emp_fac=1;
for j=1:n;

    v1(:,j,k)=emp_fac.*(Alpha(1,j).*V1(:,1)+Alpha(2,j).*V1(:,2));
    x_pert1(:,j,k)=v1(:,j,k)+x(:,k);
    x_pert1(3,j,k)=Xn(3,1);   %C soll nicht gestoert werden

    v2(:,j,k)=emp_fac.*(Alpha(1,j).*V2(:,1)+Alpha(2,j).*V2(:,2));
    x_pert2(:,j,k)=v2(:,j,k)+x(:,k);
    x_pert2(3,j,k)=Xn(3,1);

    v3(:,j,k)=emp_fac.*(Alpha(1,j).*V3(:,1)+Alpha(2,j).*V3(:,2));
    x_pert3(:,j,k)=v3(:,j,k)+x(:,k);
    x_pert3(3,j,k)=Xn(3,1);
```

Die verwendeten functions:

```
function H_norm=H_norm(X)
```

## 5 Anwendung und Implementierung

```
H_norm=zeros(num_dim);
H_norm(1,1)=sqrt(1/(4*a*b^2*pi*X(1)^2)*((1+a^2)*b^4*pi^3*(X(1)^2+X(2)^2)...
+2*pi^3*X(3)^2+4*r*b^2*f*sig*X(6)));

end

function C_norm=C_norm(X)
    C_norm=zeros(num_dim);
    C_norm(1,1)=sqrt((pi/(2*a*b*X(1)^2)*((1+a^2)*b^2*e*pi*(X(1)*X(4)...
+X(2)*X(5))+4*f*pi*X(3)*X(6)+4*X(3))));
end

function S_norm=S_norm(X)
    S_norm=zeros(num_dim);
    S_norm(1,1)=sqrt(r*sig/(12*a*pi*X(1)^2)*(3*e^2*pi*(X(4)^2+X(5)^2)...
+6*f^2*pi*X(6)^2+12*f*X(6)));
end

function norm1=norm1(X)
    norm1=H_norm(X);
end

function norm2=norm2(X)
    norm2=H_norm(X)+c1.*C_norm(X);
end

function norm3=norm3(X)
    norm3=H_norm(X)+c1.*C_norm(X)+c2.*S_norm(X);
end
```

### Ensemblevorhersage - Entwicklung der Störungen

Die Systementwicklung lässt sich schreiben als



$$\frac{d\vec{x}}{dt} = F(\vec{x}), \quad (5.21)$$

die Entwicklung des gestörten Zustandes als

$$\frac{d(\vec{x} + \vec{x}')}{dt} = F(\vec{x} + \vec{x}'), \quad (5.22)$$

wobei  $x'$  die Störung ist.

Die dazugehörige Taylorreihenentwicklung:

$$\frac{d(\vec{x} + \vec{x}')}{dt} = F(\vec{x}) + \frac{\partial F(\vec{x})}{\partial \vec{x}}|_{\vec{x}} \vec{x}' + \frac{1}{2} \frac{\partial^2 F(\vec{x})}{\partial x^2}|_{\vec{x}} \vec{x}'^2 + \dots \quad (5.23)$$

Bei einer Abschneidung nach dem 2. Term, kann für die Störungsentwicklung

$$\frac{d(\vec{x}')}{dt} = \frac{\partial F(\vec{x})}{\partial \vec{x}}|_{\vec{x}} \vec{x}' = \begin{pmatrix} \frac{\partial F_x}{\partial x} & \frac{\partial F_x}{\partial y} & \frac{\partial F_x}{\partial z} \\ \frac{\partial F_y}{\partial x} & \frac{\partial F_y}{\partial y} & \frac{\partial F_y}{\partial z} \\ \frac{\partial F_z}{\partial x} & \frac{\partial F_z}{\partial y} & \frac{\partial F_z}{\partial z} \end{pmatrix} \vec{x}' = J(\vec{x}) \vec{x}' \quad (5.24)$$

separiert werden, die Matrix  $J(\vec{x})$  wird Jacobimatrix genannt. Zu beachten ist die Linearität der Störungsentwicklung,  $J(\vec{x})$  ist ausschließlich vom Grundzustand abhängig und wird mit dem Störvektor multipliziert.

Für das System (4.36) bis (4.41) lautet die Jacobimatrix

$$J(\vec{x}) = \begin{pmatrix} 0 & \frac{a}{2b\pi(1+a^2)}(a^2-3)\pi^3 C & \frac{a}{2b\pi(1+a^2)}(a^2-3)\pi^3 B & \\ -\frac{a}{2b\pi(1+a^2)}(a^2-3)\pi^3 C & 0 & -\frac{a}{2b\pi(1+a^2)}(a^2-3)\pi^3 A & \\ 0 & 0 & 0 & \\ 0 & \frac{a\pi}{2be}(-2b^2 f \pi F - 2b^2) & \frac{a\pi}{2be} e \pi E & \dots \\ -\frac{a\pi}{2be}(-2b^2 f \pi F - 2b^2) & 0 & -\frac{a\pi}{2be}(e \pi D) & \\ -\frac{abe\pi^2}{2f} E & \frac{abe\pi^2}{2f} D & 0 & \\ & 0 & \frac{a}{2b\pi(1+a^2)} 2e R \sigma & 0 \\ & -\frac{a}{2b\pi(1+a^2)}(2e R \sigma D) & 0 & 0 \\ & 0 & 0 & 0 \\ \dots & 0 & \frac{a\pi}{2be} e \pi C & \frac{a\pi}{2be}(-2b^2 f \pi B F - 2b^2 B) \\ & -\frac{a\pi}{2be}(e \pi C) & 0 & -\frac{a\pi}{2be}(-2b^2 f \pi A) \\ & \frac{abe\pi^2}{2f} B & -\frac{abe\pi^2}{2f} A & 0 \end{pmatrix}$$

Die gestörten Modellzustände (=Analyse + Störungen (=Zufallszahl · singuläre Vektoren)) werden nun nichtlinear mit ode45 integriert.

```

for fcs=1:num_fc_steps

    sol1=ode45(@forcing,[0 fc_steps(fcs)],...
               x_pert1(:,j,k));
    x_pert_end1(:,j,k,fcs)=deval(sol1,fc_steps(fcs));
    x_pert_end1(3,j,k,fcs)=Xn(3,1);  %Rechengenauigkeit

    sol2=ode45(@forcing,[0 fc_steps(fcs)],...
               x_pert2(:,j,k));
    x_pert_end2(:,j,k,fcs)=deval(sol2,fc_steps(fcs));
    x_pert_end2(3,j,k,fcs)=Xn(3,1);

    sol3=ode45(@forcing,[0 fc_steps(fcs)],...
               x_pert3(:,j,k));
    x_pert_end3(:,j,k,fcs)=deval(sol3,fc_steps(fcs));
    x_pert_end3(3,j,k,fcs)=Xn(3,1);

end

```

Für die Berechnung der Störungen wird, wie bereits erläutert, die Annahme getroffen, dass deren Wachstum für kurze Zeit, genau genommen für die Zeitspanne bis zum nächsten Analyseschritt, linear ist. Es ist jedoch ratsam zu testen, ob diese Annahme gerechtfertigt ist, darum wird im Code die lineare mit der nichtlinearen Störungsentwicklung verglichen:

```

if k>0
    B=A;
    B(3,:)=[];
    B(:,3)=[];
    [~,~,v2test]=svd(B);
    va=[v2test(1:2,1); 0; v2test(3:5,1)];

    Xs=x(:,k)+va;

    sol=ode45(@forcing,[0 dt_ana],Xs,oo);
    Xsfc=deval(sol,0:dt:dt_ana);
    sol=ode45(@forcing,[0 dt_ana],x(:,k),oo);
    Xfc=deval(sol,0:dt:dt_ana);

```

```

diff=Xsfc-Xfc;

sprintf('Verhaeltnis:%f\n',norm(diff(:,dt_ana/dt+1))/norm(diff(:,1)))
sprintf('%d  normB %f \n',i,norm(B))
end

```

Aus der Resolvente  $B$  werden die Störungen berechnet, diese werden zum aktuellen Zustand addiert und es wird sowohl der Zustand als auch der gestörte Zustand mittels Runge-Kutta-Verfahren integriert. Das nichtlineare Störungswachstum lässt sich aus der Differenz dieser beiden durch Betrachtung der Anfangs- und der Endnorm ermitteln. Das lineare Wachstum ergibt sich aus der Norm der Resolvente.

Zu erwähnen bleibt noch die in Kap. 2.2 auf Seite 25 erwähnte Lyapunov Analyse um das deterministische Chaos des erweiterten Lorenz(1963)-Modells mathematisch nachzuweisen:

```

for i=1:(t_end/dt);

    ...

    jac=jacobi(Xn(:,i));

    for m=1:num_dim
        Lyapunov(m,i)=log(norm(jac(m,:)));
    end

    ...

end

...

for m=1:num_dim
    Lyapunov_spec(m,1)=sum(Lyapunov(m,:))/i;
end

fprintf('Lyapunov Spektrum:\n');
fprintf('%f\n',Lyapunov_spec);

```

#### 5.2.2.4 Verifikation

Da in den Zeitreihen die einzelnen Komponenten aperiodisch zwischen positiven und negativen Werten schwanken, wird zur Verifikation eine Einteilung in  $x_i > 0$  (=Beobachtung-ja) und  $x_i \leq 0$  (Beobachtung-nein) vorgenommen.

Weiters werden im Code die Zeitreihen so zugeschnitten, dass für alle Vorhersageschritte die maximale, für alle verfügbare Reihe verifiziert wird.

Der Root Mean Sqare Error (rms) und der Ensemblespread werden verglichen, um ihre Korrelation zu testen. Das EPS sollte sensible Bereiche erkennen und an diesen einen höheren Ensemblespread zur Verfügung stellen.

#### Root Mean Sqare Error

Für den Vergleich wird die Zeitreihe der quadratischen Abweichungen verwendet

$$rms(k) = (\vec{x}_{fc(k)} - \vec{x}_{(k)})^2 \quad (5.25)$$

wobei  $\vec{x}_{fc}$  die Vorhersagen des nichtlinearen Modells und  $\vec{x}$  die Analysen nach Verwendung des Kalmanfilters sind.

#### Ensemblespread

Die Zeitreihe des Ensemblespreads ist

$$spread(k) = \frac{1}{m-1} \sum_{m=1}^M (\vec{x}_{fc(k,m)} - \bar{\vec{x}}_{fc(k)})^2 \quad (5.26)$$

wobei  $M$  die Anzahl der Ensembleläufe,  $\vec{x}_{fc(k,m)}$  jede einzelne Vorhersage eines Ensemblemitglieds und  $\bar{\vec{x}}_{fc(k)}$  das Ensemblemittel zum Zeitpunkt  $k$  ist. Der Multiplikator  $\frac{1}{m-1}$  resultiert aus der Reduktion der Freiheitsgrade durch Verwendung des Ensemblemittels.

#### Brier Score

Zu Beginn eine Tabelle, die alle vorkommenden Größen zusammenfasst. Als Beobachtungen werden die Analysen verwendet, diese sollten definitionsgemäß näher an der Wahrheit liegen, als die unbearbeiteten Beobachtungen.

## 5 Anwendung und Implementierung

Parameter	Bedeutung	Wertebereich
$n$	Anzahl der Vorhersagen insgesamt	$\mathbb{N}^+$
$p_k$	lt. Ensemblemitgliedern vorhergesagte Wahrscheinlichkeit	$[0, 1]$
$o_k, o_l$	Beobachtung-ja / Beobachtung-nein (aus Analyse)	$\{0, 1\}$
$p_i$	lt. Ensemblemitgliedern vorhergesagte Wahrscheinlichkeit, eingeteilt in Klassen	$[0, 10\%], \dots$
$I$	Anzahl der Vorhersageklassen $p_i$	$\mathbb{N}^+$
$N_i$	Anzahl der Vorhersagen in bestimmter Klasse $p_i$	$\mathbb{N}^+$
$\bar{o}_i$	Bedingte Wahrscheinlichkeit für gegebene vorhergesagte Klasse	$[0, 1]$
$\bar{o}$	a priori Wahrscheinlichkeit, dass Beobachtung eintritt	$[0, 1]$

Tabelle 5.3: Größen zur Verifikation

wobei

$$\bar{o}_i = \frac{1}{N_i} \sum_{l=1}^{N_i} o_l \quad (5.27)$$

und

$$\bar{o} = \frac{1}{n} \sum_{k=1}^n o_k \quad (5.28)$$

Der Brier Score zeigt den quadratischen Abstand zwischen der vorhergesagten Wahrscheinlichkeit (= % der Ensembleläufe, die Event-ja vorhersagen) und der upgedateten Beobachtungen, (=Analysen), (Event-ja  $\mapsto o_k = 1$ ):

$$BS = \frac{1}{n} \sum_{k=1}^n (p_k - o_k)^2 \quad (5.29)$$

Der Brier Score kann in Zuverlässigkeit (reliability), Auflösung (resolution) und Unsicherheit (uncertainty) aufgespalten werden.

Zuverlässigkeit:

$$BS_{rel} = \frac{1}{n} \sum_{i=1}^I N_i (p_i - \bar{o}_i)^2 \quad (5.30)$$

Auflösung:

$$BS_{res} = \frac{1}{n} \sum_{i=1}^I N_i (\bar{o}_i - \bar{o})^2 \quad (5.31)$$

Unsicherheit:

$$BS_{unc} = \bar{o} (1 - \bar{o}) \quad (5.32)$$

Wie aus den Formeln ersichtlich, sollte der  $BS$  möglichst nahe bei 0 liegen, auch  $BS_{rel}$  sollt klein sein. Die Auflösung ist umso besser je größer  $BS_{res}$ , für kleine Werte liegt die vorhergesagte Wahrscheinlichkeit nahe der 50% und bringt nicht wirklich viel Zusatzinformation zu einer Zufallsprognose.

$BS_{unc}$  zeigt, wie schwierig eine Größe generell vorherzusagen ist. Die Herausforderung ist am größten für ein statistisches Auftreten der Beobachtung um 50%,  $\bar{o} = 0.5$  und  $BS_{unc} = 0.25$ . Dieser Wert wird in den Ergebnisanalysen nicht besprochen, da er unabhängig der Berechnungen ist und somit für den Vergleich der Normen irrelevant.

### Relative Operation Characteristics

Beim Relative Operation Characteristics (ROC) - Diagramm werden auf der  $x$ -Achse die POFD und auf der  $y$ -Achse die POD aufgetragen, welche sich folgendermaßen berechnen:

	beobachtet <sup>6</sup>	nicht beobachtet
vorhergesagt	H (hits)	F (false)
nicht vorhergesagt	M (missed)	Z (zero)

Tabelle 5.4: Kontingenztafel

Die Bezeichnung „beobachtet“ in der Kontingenztafel bezieht sich auf die upgedatete Beobachtung, also die Analyse.

Erkennungswahrscheinlichkeit, Probability of Detection - POD:

$$POD = \frac{H}{H + M} \quad (5.33)$$

Falsche Alarmrate, Probability of false Detection - POFD:

$$POFD = \frac{F}{F + Z} \quad (5.34)$$

Es wird also die Wahrscheinlichkeit ein Event richtig vorherzusagen mit der Wahrscheinlichkeit es fälschlicherweise vorherzusagen verglichen.

#### 5.2.2.5 Plots

Geplottet werden unter anderem:

- 2d Schnitte des Phasenraums mit Attraktor
- 3d Schnitte des Phasenraums mit Attraktor
- Temperatur- und Vorticityfelder
- Entwickelte Komponenten mit Störungen und Ensembleberechnungen (verschiedene Normen)
- Zeitliche Entwicklung der konservativen Größen zur Überprüfung der numerischen Erhaltung, absolut und normiert auf den Startwert
- Spur der Fehlerkovarianzmatrix  $P$  zur Stabilitätsüberprüfung
- Darstellung der generierten Beobachtungen
- Überprüfung der Phasenverschiebung zwischen freier Vorhersage und Vorhersage mit Update-Schritten
- Gegenüberstellungen von Beobachtungen, Vorhersagen und Analysen zur Überprüfung einer adäquaten Berücksichtigung der ersten beiden zur Berechnung der letzteren
- Darstellung der Winkel zwischen den singulären Vektoren, die durch die verschiedenen Normen berechnet wurden
- Vergleich von rms, Ensemblespread und der Spur von  $P$ , je für die verschiedenen Normen
- Reliability Diagramme der einzelnen Normen
- ROC-Diagramme der einzelnen Normen

Die relevanten Plots werden im Kap. 6 dargestellt.

#### 5.2.2.6 Functions

Folgende functions werden definiert, der zugehörige Programmcode befindet sich, wenn relevant, jeweils bei der Beschreibung der Verwendung.

- forcing - zur Systementwicklung
- jacobi - zur Entwicklung der Störungen
- H\_Norm - zur Berechnung der singulären Vektoren bezüglich der Energie (Hamilton)-Norm

## 5 Anwendung und Implementierung

- C\_Norm - zur Berechnung der singulären Vektoren bezüglich der Helizitäts-Norm
- S\_Norm - zur Berechnung der singulären Vektoren bezüglich der Temperatur-Norm
- angle - zur Berechnung der Winkel zwischen den verschiedenen singulären Vektoren
- norm1 - reine Energienorm
- norm2 - Kombination aus Energie- und Vorticity-Norm
- norm3 - Kombination aus Energie-, Vorticity- und Temperatur-Norm
- H\_norm\_cons\_check - zur Überprüfung, ob Energie während Systemintegration tatsächlich erhalten bleibt
- C\_norm\_cons\_check - zur Überprüfung, ob Helizität während Systemintegration tatsächlich erhalten bleibt
- S\_norm\_cons\_check - zur Überprüfung, ob Temperaturkonturen während Systemintegration tatsächlich erhalten bleibt



## 6 Ergebnisse

Die Quelle aller folgenden Plots sind die Codes auf Seite 189 und 229.

Da das implementierte Modell noch eher unerforscht ist, werden zunächst einige allgemeine Betrachtungen diskutiert. Dafür werden die Standardeinstellungen (siehe Kap. 5.2.2.1) verwendet, insbesondere:

```
Xn=[1 1 1 1 1 1]  
b=1
```

Diese Einstellungen schaffen in gewisser Weise symmetrische Verhältnisse, die dynamischen Variablen bewegen sich in gleichen Größenordnungen und die beiden geometrischen Faktoren  $a$  und  $b$  sind beide 1.

Für die Bewertung des EPS in Kap. 6.7 und 6.8 durch die verschiedenen Normen werden die in Kap. 5.2.2.3 eingeführten, leicht abgeänderten Parameter benutzt:

```
Xn=[1 1 4 1 1 1]  
b=1.102
```

Diese Einstellungen bewirken, dass die Diskriminanten der verwendeten Normen nicht negativ werden und somit die Singulärwertzerlegung zur Störungsgenerierung nicht durch Imaginärteile verfälscht wird. Wie in Kap. 5.2.2.3 beschrieben, ist dies eine Grundvoraussetzung um die Erhaltungseigenschaften des Systems zur Optimierung der Ensemblevorhersage einsetzen zu können.

Sämtliche Überprüfungen, die Funktion des Modells betreffend, wurden für beide Einstellungen durchgeführt und zeigen im Allgemeinen ähnliche Verhaltensweisen.

Die allgemeinen Darstellungen vor den Untersuchungen bezüglich der eingeführten Normen in Kap. 6.7 erfolgen über die Standardeinstellungen, wenn nicht wegen deutlichen Unterschieden explizit anders angegeben.

## 6.1 Lyapunov Analyse

Die Systemanalyse nach Lyapunov wie in Kap. 2.2 auf Seite 25 eingeführt, bringt für den freien Lauf über 100 Zeitschritte folgendes Spektrum:

3.193166  
3.193430  
-Inf  
3.104583  
3.105598  
2.758257

Damit ist das deterministische Chaos des Modells eindeutig nachgewiesen.

Die Tatsache, dass die Exponenten der erweiterten Lorenz-Gleichungen signifikant größer als die der Originalgleichungen von 1963 (siehe 25) sind, deutet schon auf eine höhere Sensibilität hin, welche in Kap. 6.5 noch näher untersucht wird.

## 6.2 Phasenraumplots

Zwar ist es bei einem 6-dimensionalen System nicht möglich sich den vollständigen Attraktor räumlich vorzustellen, jedoch lassen sich relativ viele Unterphasenräume darstellen.

Zunächst wurde bis  $t_{\text{end}}=5$  integriert um die einzelnen Trajektorien noch sichtbar zu halten, in Abb. 6.1<sup>1</sup> und Abb. 6.2<sup>2</sup> zu sehen sind alle möglichen 2- bzw. 3-dimensionalen Phasenraumschnitte:

---

<sup>1</sup>Quelle: Code auf Seite 182

<sup>2</sup>Quelle: Code auf Seite 182

## 6 Ergebnisse

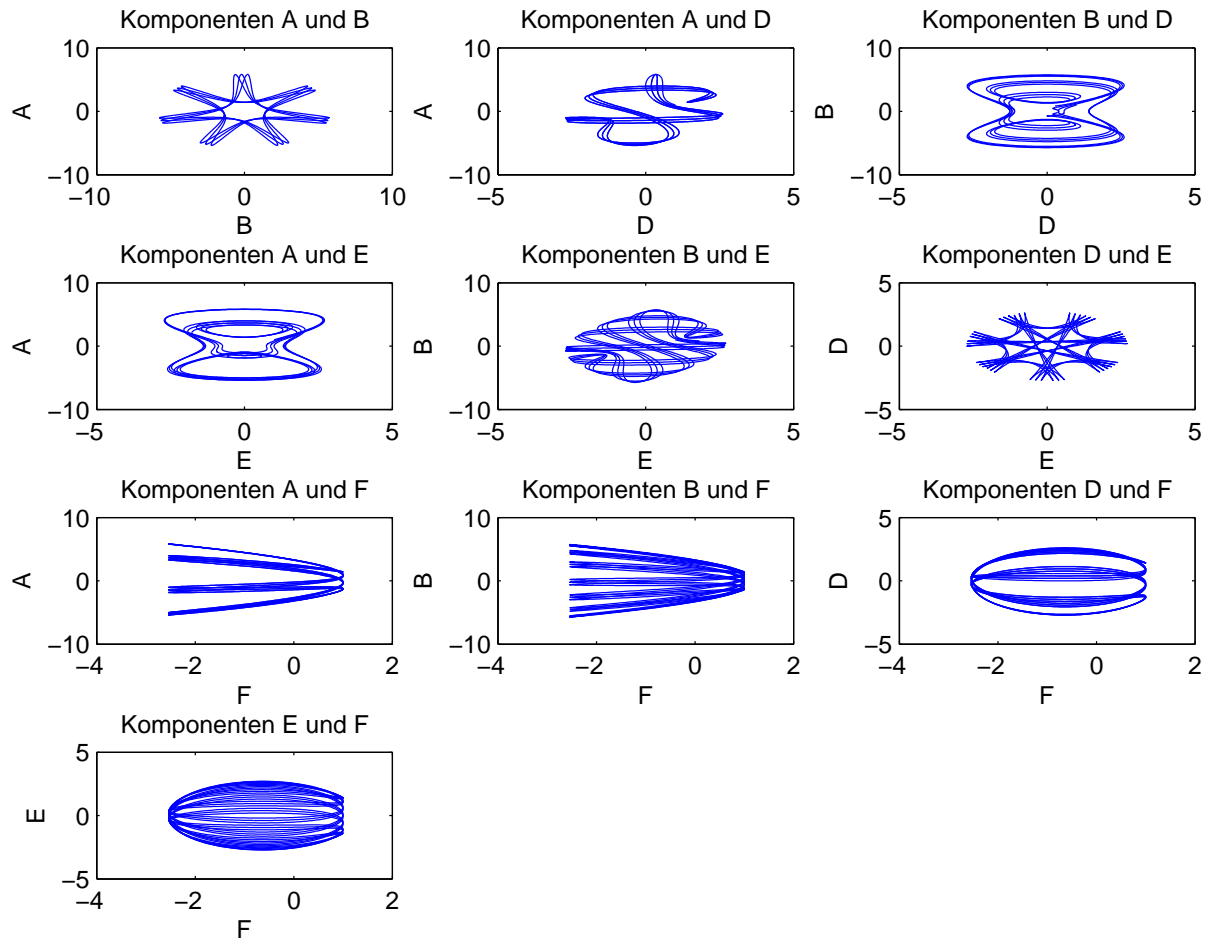


Abbildung 6.1: 2d Phasenraumschnitte,  $t_{\text{end}}=5$

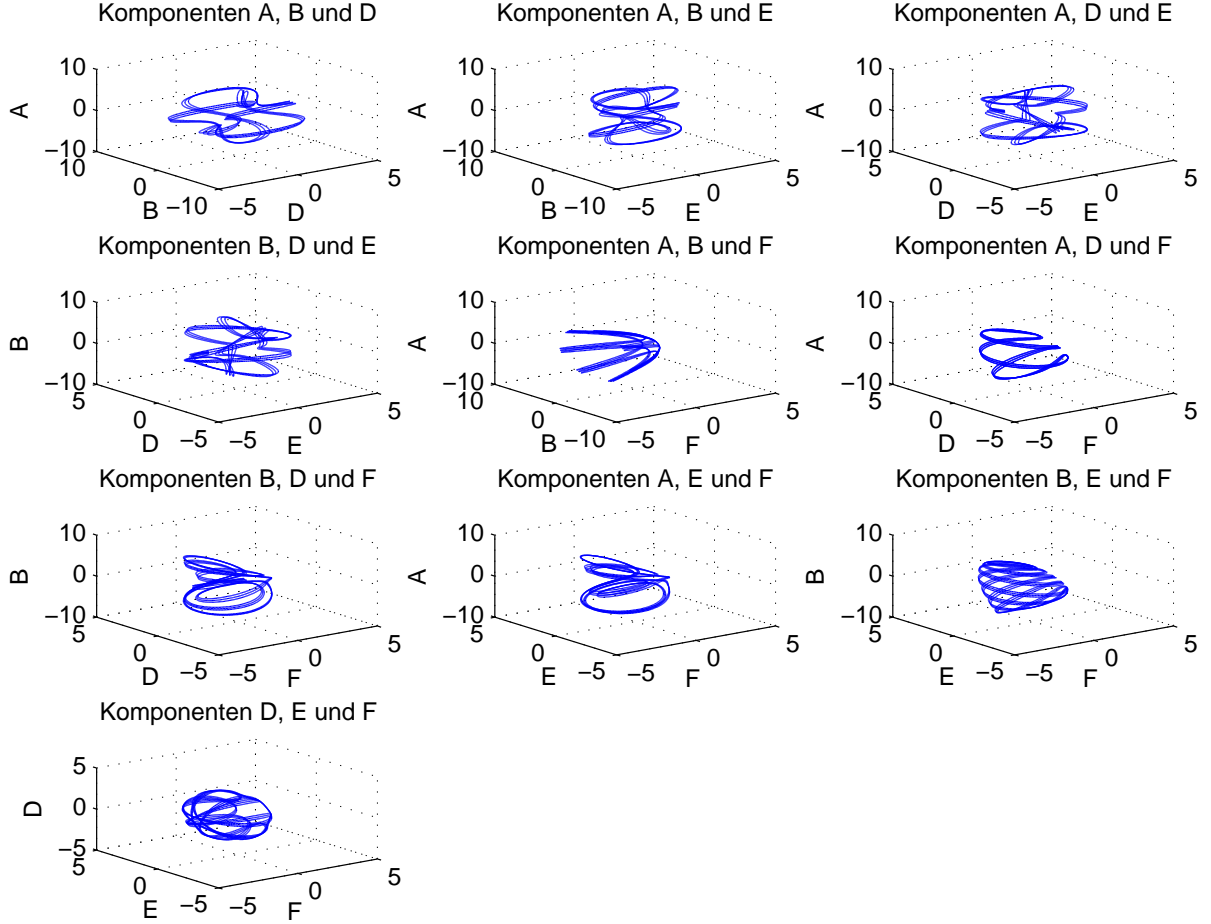


Abbildung 6.2: 3d Phasenraumschnitte,  $t_{\text{end}}=5$

Mit etwas Phantasie lassen sich in manche dieser 3-dimensionalen Teilattraktoren erweiterte Achterschleifen, ähnlich dem 3-dimensionalen Modell interpretieren, an dieser Stelle soll die physikalische Bedeutung der Komponenten  $A - F$ , im Vergleich zu  $X, Y, Z$  genauer betrachtet werden.

Nachdem Lorenz in [23] unter Verwendung der numerischen Lösungen Barry Saltzmans das 3-dimensionale Modell hergeleitet hat, schreibt er: “The same solutions would have been obtained if the series had at the start been truncated to include a total of three terms. Accordingly, in this study we shall let

$$\frac{1}{a(1+a^2)\kappa}\Psi = X\sqrt{2}\sin\left(\frac{\pi a}{H}x\right)\sin\left(\frac{\pi}{H}z\right) \quad (6.1)$$

$$\frac{\pi R_a}{R_c \Delta T}\theta = Y\sqrt{2}\cos\left(\frac{\pi a}{H}x\right)\sin\left(\frac{\pi}{H}z\right) - Z\sin\left(\frac{2\pi}{H}z\right) \quad (6.2)$$

where  $X, Y$  and  $Z$  are functions of time alone.“

Wie auch schon in Kap. 2.2 beschrieben, ist hier deutlich zu sehen, dass  $X$  die 2-dimensionale Strömung mit einem geometrischen Faktor  $a$  beschreibt.  $Y$  zeigt die cosinusförmige horizontale Temperaturabweichung einer homogenen Verteilung, die sich mit der Höhe  $z$  sinusförmig ändert.  $Z$  ist der Temperaturabweichung im vertikalen Profil zugehörig.

Stellt sich die Frage, ob sich diese horizontal nicht ändert bzw. ob das Modell diese Änderung nicht erfassen kann.

Für die Berücksichtigung von sechs Fourierkoeffizienten wurden die Felder in (4.25) und (4.26) angeschrieben:

$$\Psi = \sum_{n=0}^{\infty} \sum_{m=1}^{\infty} (\phi_{nm} \sin an\pi x + \varphi_{nm} \cos an\pi x) \sin m\pi z \quad (6.3)$$

und

$$T = \sum_{n=0}^{\infty} \sum_{m=1}^{\infty} (\vartheta_{nm} \sin an\pi x + \theta_{nm} \cos an\pi x) \sin m\pi z \quad (6.4)$$

Es werden sowohl zur Beschreibung des Strömungs- als auch des Temperaturfeldes je zwei verschiedene Koeffizienten verwendet. Nach der Wahl der Moden, also der Konkretisierung von  $n$  und  $m$ , bleiben je 3 Entwicklungskoeffizienten für jedes Feld übrig. Da genau diese Koeffizienten in skalierten Form bereits die Systemvariablen  $A$  bis  $F$  sind, kann wiederum eine direkte Zugehörigkeit angegeben werden.

Aus  $\phi_{11} = bA$ ,  $\varphi_{11} = bB$ ,  $\vartheta_{11} = eD$ ,  $\theta_{11} = eE$ ,  $\varphi_{02} = cC$ ,  $\theta_{02} = fF$  wird ersichtlich, dass  $A, B$  und  $C$  für das Strömungsfeld verantwortlich sind,  $D, E$  und  $F$  Temperaturabweichungen vertreten. Die Darstellung scheint in 6 Dimensionen weitaus konsistenter, sowohl Strömungs- als auch Temperaturabweichungen werden horizontal aus einer Sinus- und zwei Cosinusfunktionen kombiniert, die sich mit der Höhe sinusförmig ändern.  $C$  liefert einen konstant gewichteten Cosinusbeitrag zum Strömungsfeld, dieses hat demnach anscheinend einen Freiheitsgrad weniger in seiner Entwicklung.

Wird zeitlich noch etwas weiter integriert, so wird in Abb. 6.3 und Abb. 6.4 deutlich, was bereits Abb. 6.1 und Abb. 6.2 vermuten ließen:

## 6 Ergebnisse

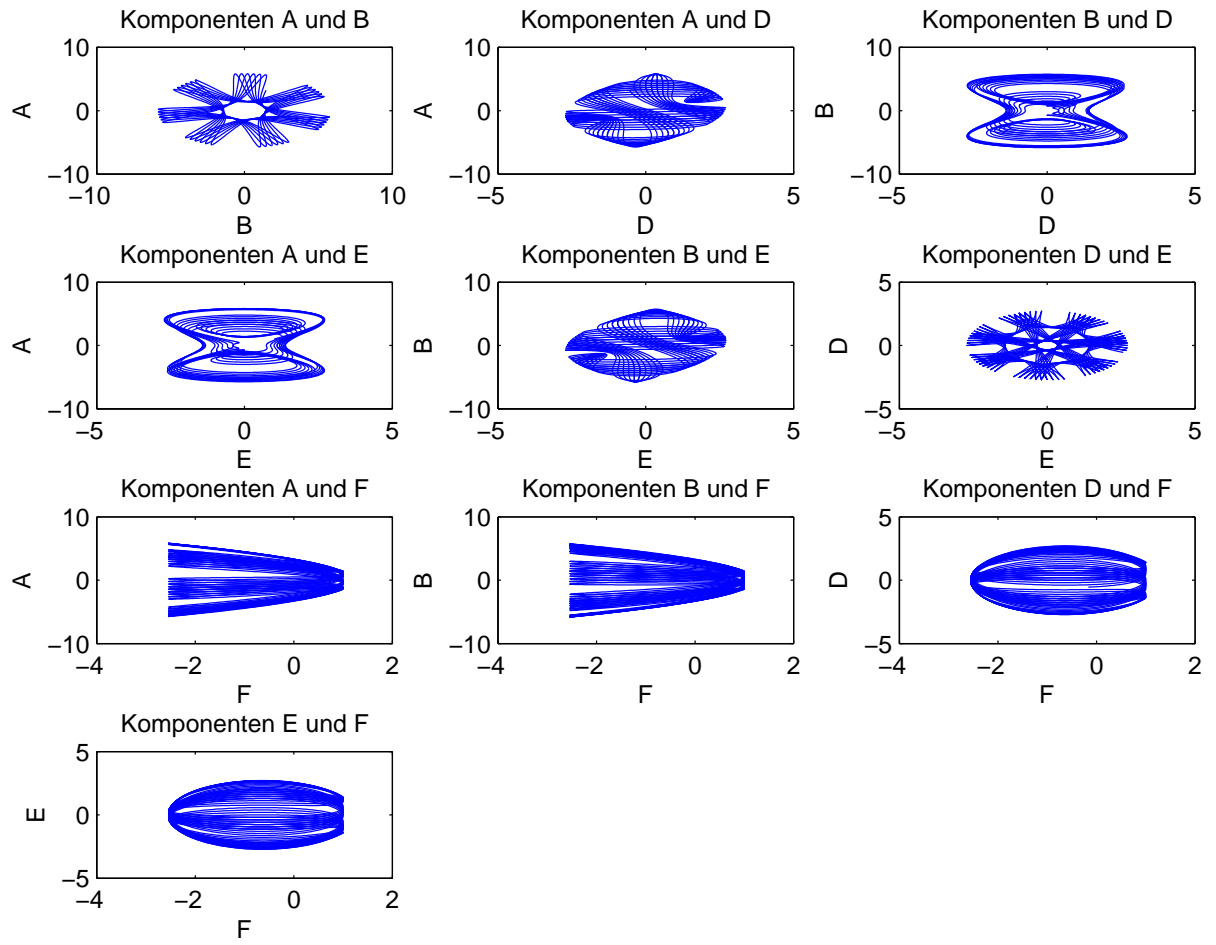


Abbildung 6.3: 2d Phasenraumschnitte,  $t_{\text{end}}=10$

## 6 Ergebnisse

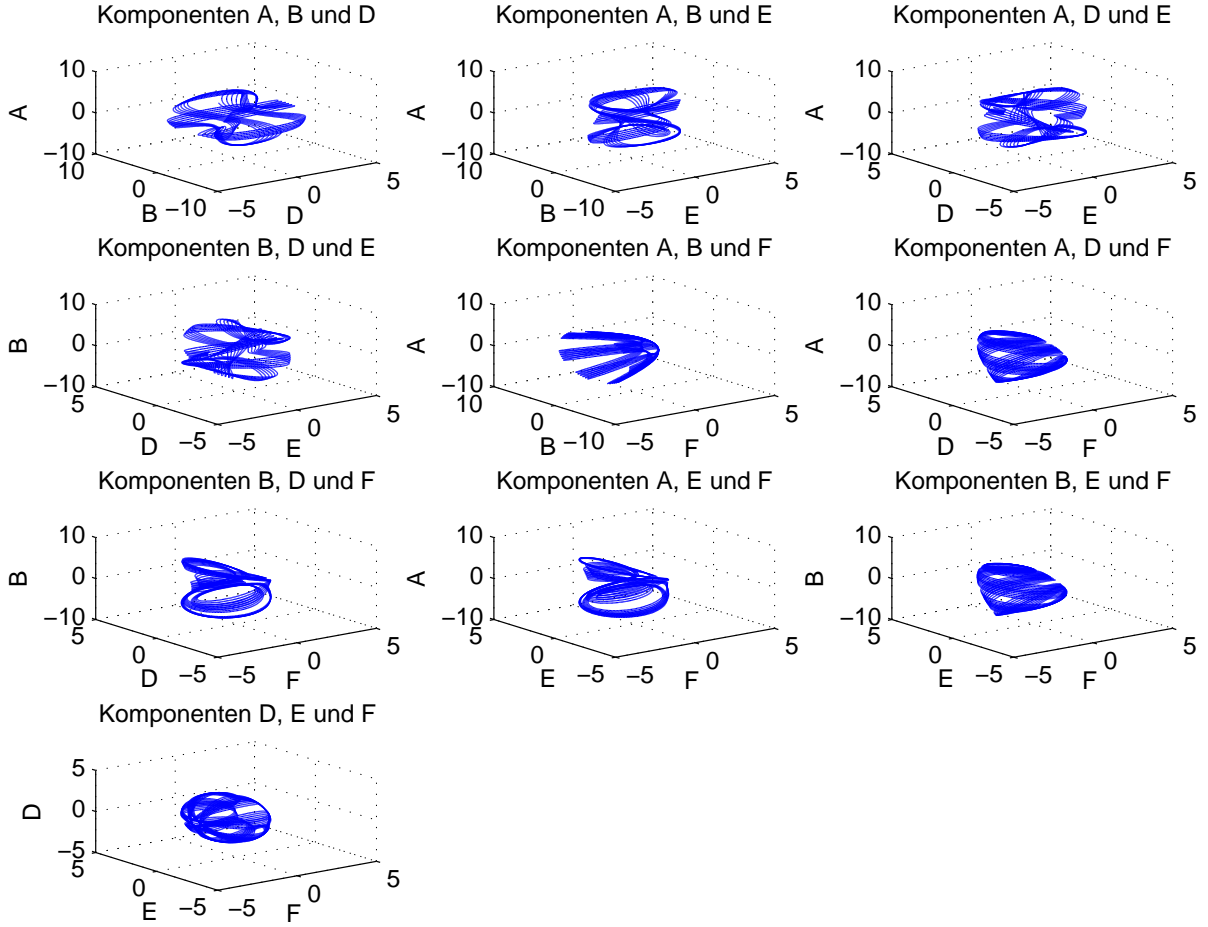


Abbildung 6.4: 3d Phasenraumschnitte,  $t_{\text{end}}=10$

Die 2-dimensionalen Phasenraumschnitte von zum Beispiel  $AF$  und  $BF$  sowie von  $DF$  und  $EF$  bilden nach einiger Zeit sehr ähnliche Attraktoren. Demzufolge entwickeln sich  $A$  und  $B$  bzw.  $D$  und  $E$  auf ähnlichen Bahnen, allerdings vermutlich in verschiedene Richtungen. Diese Vermutung beruht auf der Antisymmetrie der Entwicklungsgleichungen, die hier noch einmal angeschrieben werden:

$$\frac{dA}{dt} = \frac{a}{2b\pi(1+a^2)} \left( (a^2 - 3) \pi^3 BC + 2eR\sigma E \right) \quad (6.5)$$

$$\frac{dB}{dt} = -\frac{a}{2b\pi(1+a^2)} \left( (a^2 - 3) \pi^3 AC + 2eR\sigma D \right) \quad (6.6)$$

$$\frac{dC}{dt} = 0 \quad (6.7)$$

$$\frac{dD}{dt} = \frac{a\pi}{2be} \left( e\pi CE - 2b^2 f\pi BF - 2b^2 B \right) \quad (6.8)$$

$$\frac{dE}{dt} = -\frac{a\pi}{2be} (e\pi CD - 2b^2 f\pi AF - 2b^2 A) \quad (6.9)$$

$$\frac{dF}{dt} = \frac{abe\pi^2}{2f} (BD - AE) \quad (6.10)$$

Jedoch ist aufgrund der Nichtlinearität des Modells auf den ersten Blick nicht zu sagen, was nach dem 1. Integrationsschritt geschieht, wahrscheinlich unterscheiden sich die korrelierenden Größen auch bezüglich der Geschwindigkeit ihrer Trajektorien im Phasenraum.

Auf jeden Fall weisen die ähnlich verlaufenden dynamischen Variablen  $A$  und  $B$  sowie  $D$  und  $E$  auf eine gewisse Regelmäßigkeit in der Entwicklung des Strömungs- bzw. Temperaturfeldes trotz chaotischem Charakter des Modells hin.

Mit diesen Einschränkungen der Komponenten stellt sich auch die interessante Frage nach der Dimensionalität des gesamten Attraktors, die, wie die des 2,06-dimensionalen Attraktors des 3-dimensionalen Modells, wahrscheinlich auch nicht ganzzahlig ist, vor allem wenn auch der dissipative Teil des Systems betrachtet wird.

Wenn im konservativen Modell  $A$  wirklich  $B$  und  $D$  wirklich  $E$  entspricht (nur rückwärts und mit unterschiedlich schnellen Trajektorien) und  $C$  konstant ist, vielleicht kann der Attraktor dann sogar 3-dimensional aus z.B.  $A, D$  und  $F$  dargestellt werden? Kann das 6-dimensionale Modell unter Umständen einen ganzzahligen Attraktor für seinen konservativen Kern bewirken?

Durch die in Kapitel 5.2.2.3 ermittelten Parameter kann die Komponente  $F$  nicht nur positiv gehalten werden, sie schwankt auch nur noch unerheblich zwischen 0.95 und 1. Wenn  $F$  durch dieses fast konstante Verhalten nun als Gegenstück zu  $C$  interpretiert wird, was mag dann weiter mit den Attraktordimensionen geschehen?

Soweit nur einige spontane Gedanken zum untersuchten Modell, diese sollen nun aber wieder zugunsten der weiteren, konkreten Modellanalyse verworfen werden.

## 6.3 Zeitreihe der Komponenten

In Abb. 6.5<sup>3</sup> sind die entwickelten Zeitreihen aller Komponenten im Überblick geplottet. Es ist das Grundkonzept der Ensemblevorhersage zu erkennen, zu jedem Analyseschritt werden Störungen generiert und diese entwickelt (linear approximierte Störungsentwicklung in grün). Je nach Modellunsicherheit weisen die entwickelten Störungen größere oder kleinere Auffächerungen auf.

---

<sup>3</sup>Quelle: Code auf Seite 189



## 6 Ergebnisse

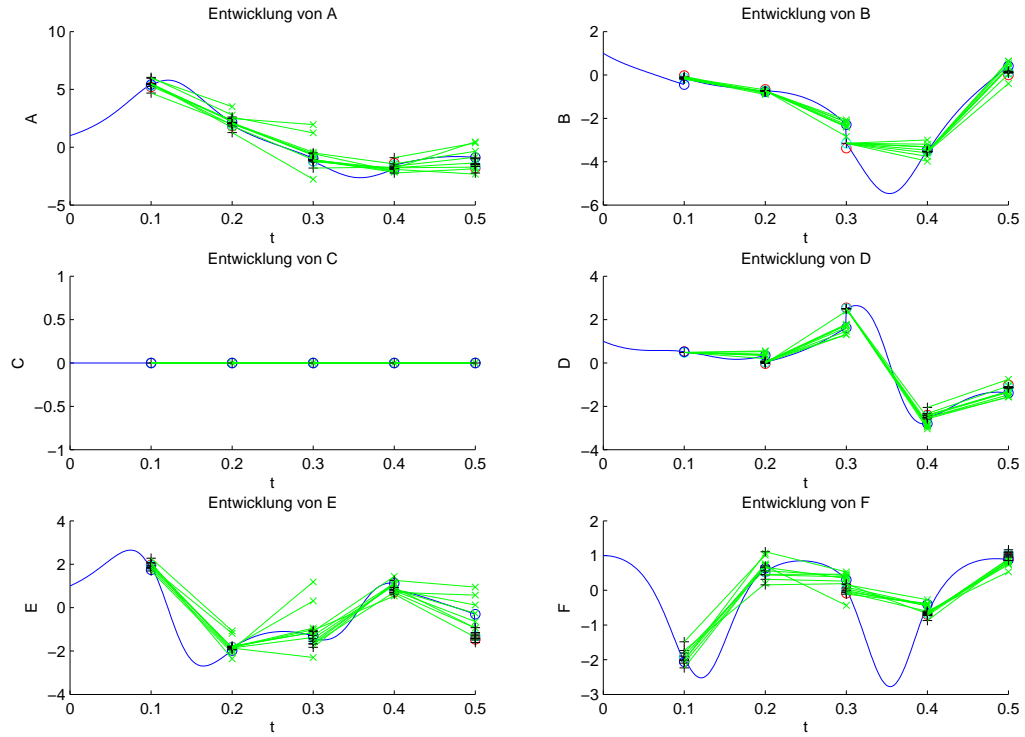


Abbildung 6.5: Übersicht: Zeitliche Entwicklung der Komponenten  $A - F$

Für genauere Betrachtungen soll in Abb. 6.6<sup>4</sup> nur die erste Komponente dargestellt werden:

<sup>4</sup>Quelle: Code auf Seite 189

## 6 Ergebnisse

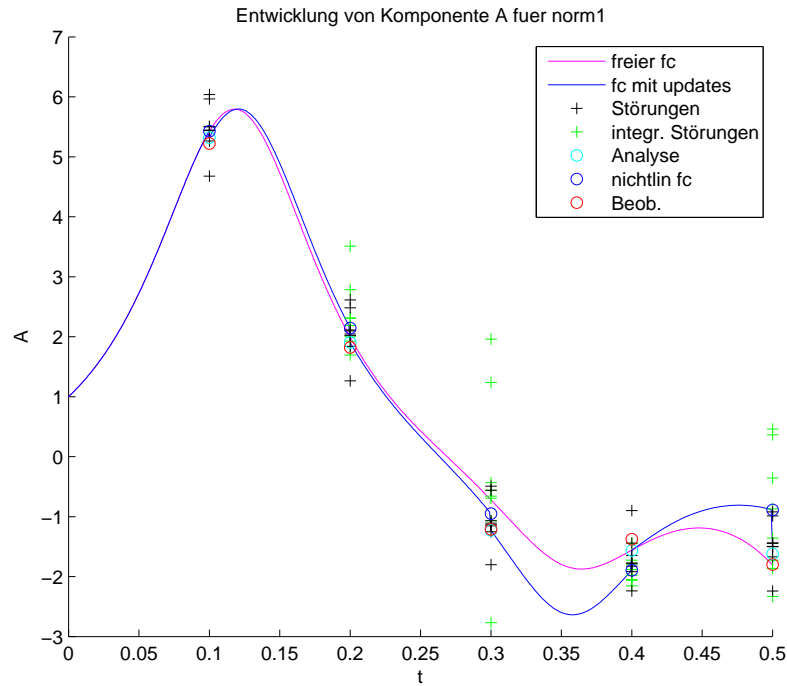
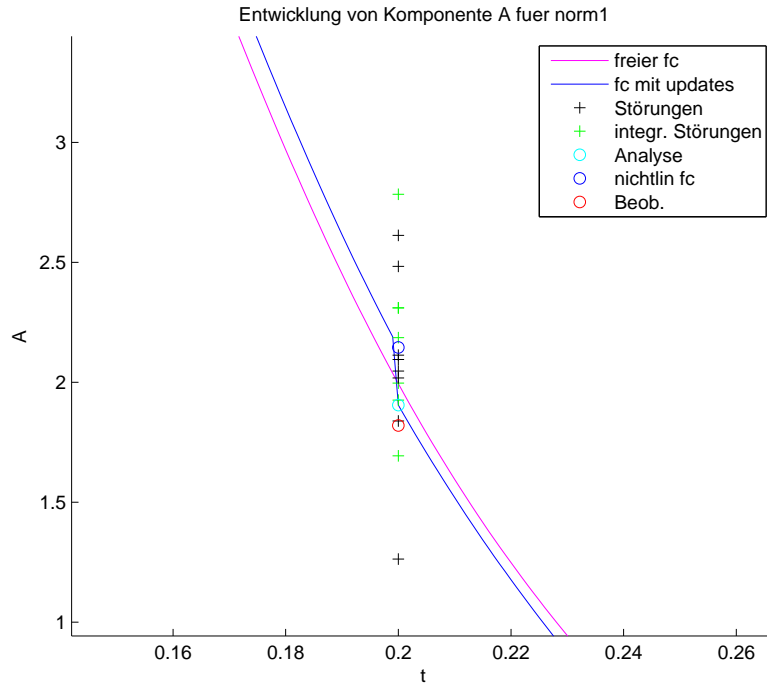


Abbildung 6.6: Zeitliche Entwicklung der Komponente  $A$

In Magenta ist die freie Vorhersage zu sehen, auf der auch die Beobachtungen (rot) basieren. Die blaue Trajektorie zeigt die zeitliche Entwicklung von  $A$  unter Verwendung der Analysen und Updates. Bei  $t = 0.1$  werden die ersten Störungen generiert und bis  $t = 0.2$  entwickelt.

Der Zeitpunkt  $t = 0.2$  wird nun noch vergrößert um zu zeigen, was geschieht:

Abbildung 6.7: Zeitliche Entwicklung der Komponente  $A$  - Zoom

Die grünen Kreuze zeigen die entwickelten Störungen. Im Bereich, den diese einschließen, sollte die tatsächlich eingetretene Beobachtung liegen, dann ist die Ensemblevorhersage hier erfolgreich. Dies ist tatsächlich der Fall, in rot ist die Beobachtung dargestellt. Der blaue Kreis zeigt die Vorhersage des nichtlinearen Modells, nach Verwendung des Kalmanfilters ergibt sich aus Beobachtung und Vorhersage durch ein gewichtetes Mittel die Analyse, hier als türkiser Kreis. Durch den Update-Schritt wird die Trajektorie von  $A$  (in blau) sofort angepasst, die neue Information wird in das System eingebunden. Von diesem Wert ausgehend werden wieder Störungen generiert (schwarze Kreuze) und bis zum nächsten Zeitschritt integriert.

## 6.4 Temperatur- und Vorticityfelder

Um den Zusammenhang zwischen dem Modell im spektralen Raum und den realen Abläufen im physikalischen Raum zu finden, können die Strömungs- und Temperaturfelder, die die Entwicklungskoeffizienten gemäß Gl. (6.3) und (4.26) liefern, berechnet werden.

Eine solche Darstellung zeigen Abb. 6.8 und Abb. 6.9 <sup>5</sup>

---

<sup>5</sup>Quelle: Code Quelle: auf Seite 189

## 6 Ergebnisse

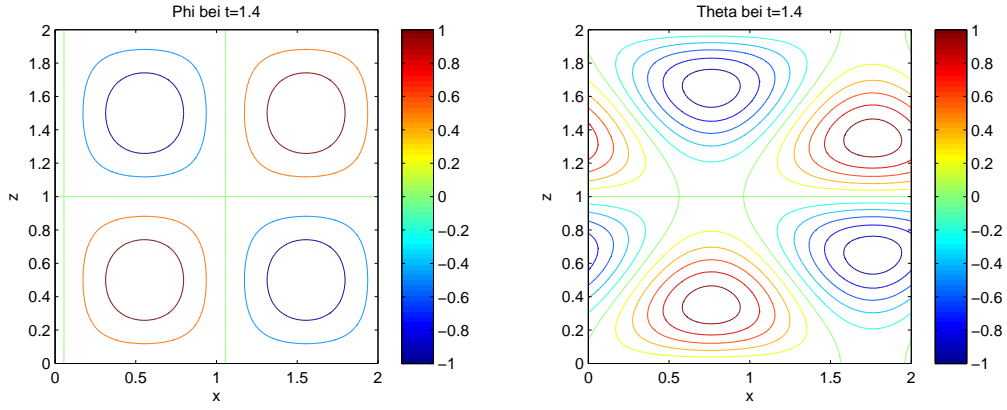


Abbildung 6.8: Strömungs- und Temperaturfeld  $t = 1.4$

Zu erkennen ist das vertraute Bild eines temperatursausgleichenden Strömungsmusters. Im linken unteren Quadranten befindet sich eine zyklonale Konvektionsrolle, identifizierbar durch die positiven Vorticitywerte, deren Strömung im Bereich  $x = 0, 0.5$  abwärts und  $x = 0.5, 1$  aufwärts gerichtet ist. Im Temperaturfeld ist in etwa bei  $x = 0.75$  das Maximum einer positiven Temperaturstörung auszumachen, die Strömung ist also bemüht diese aufwärts zu transportieren. Gleichzeitig korreliert die Abwärtsströmung der antizyklonalen Konvektionsrolle im rechten unteren Quadranten mit dem Temperaturminimum bei  $x = 1.75$ .

Nach kurzer zeitlicher Entwicklung bringt eine Momentaufnahme folgenden Zustand:

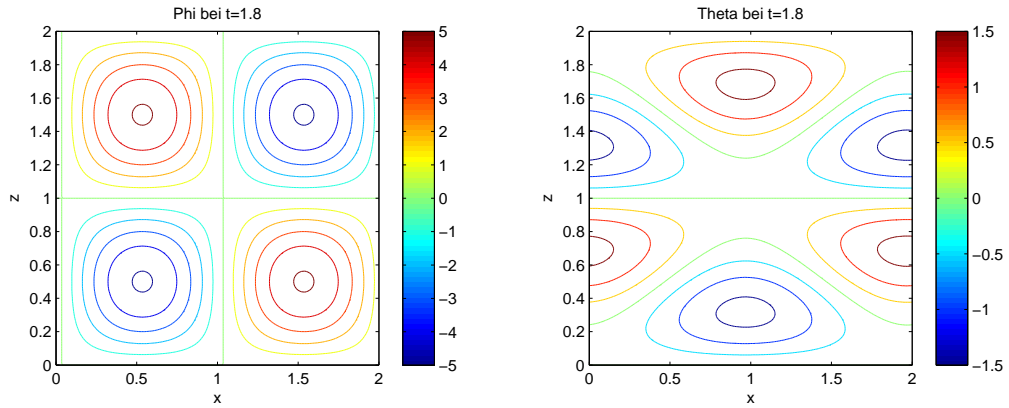


Abbildung 6.9: Strömungs- und Temperaturfeld  $t = 1.8$

Um  $x = 1$  sammelt sich relativ zur Umgebung kälteres Fluid am Boden des Gefäßes an, hier verläuft die Nulllinie des Strömungsmusters, das Volumen verharrt in Ruhe. Rechts und links außen werden wärmere Fluidmassen durch die Zyklone rechts und die Antizyklone links aufwärts beschleunigt, im oberen Abschnitt lässt sich das Gegenteil der gesamte Situation beobachten.

## 6.5 Stabilität

Bei der Implementierung der Systemgleichungen ist es wichtig, dass diese nicht durch das numerische Integrationschema, zu große Integrationschritte usw. instabil werden. Aus diesem Grund wird das System zu aller erst frei, also ohne Beobachtungen, Analysen oder Störungen über eine längere Zeitspanne in Abb. 6.10<sup>6</sup> entwickelt.

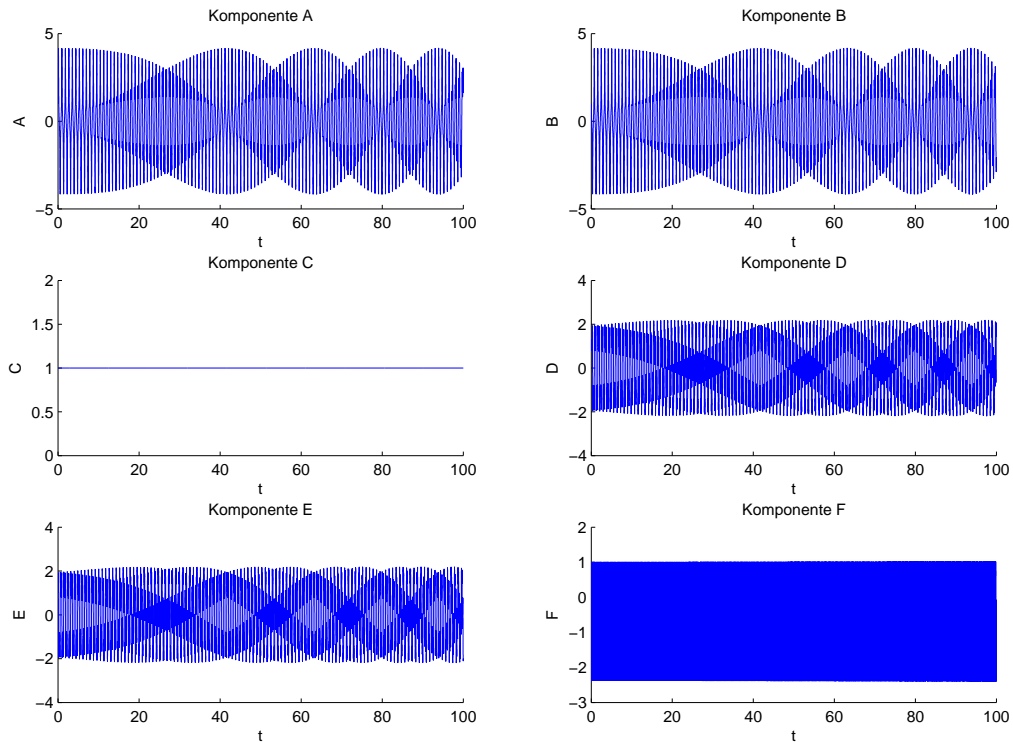


Abbildung 6.10: Freie Entwicklung der Komponenten  $A - F$

Alle Komponenten bleiben bei dem gewählten Integrationschritt für das Heun-Schema mit einem Zwischenschritt stabil, das gewählte  $dt=0.001$  kann also für alle weiteren Berechnungen verwendet werden.

Auch an dieser Stelle fällt auf, dass  $A$  und  $B$ , sowie  $D$  und  $E$  sehr ähnliche Zeitreihen zeigen, dies unterstützt die physikalische Interpretation der Komponenten in Kap. 6.2.

Da Erhaltungsgrößen eine große Rolle spielen, sollte auch kontrolliert werden, ob diese tatsächlich erhalten bleiben.

---

<sup>6</sup>Quelle: Code auf Seite 186

## 6 Ergebnisse

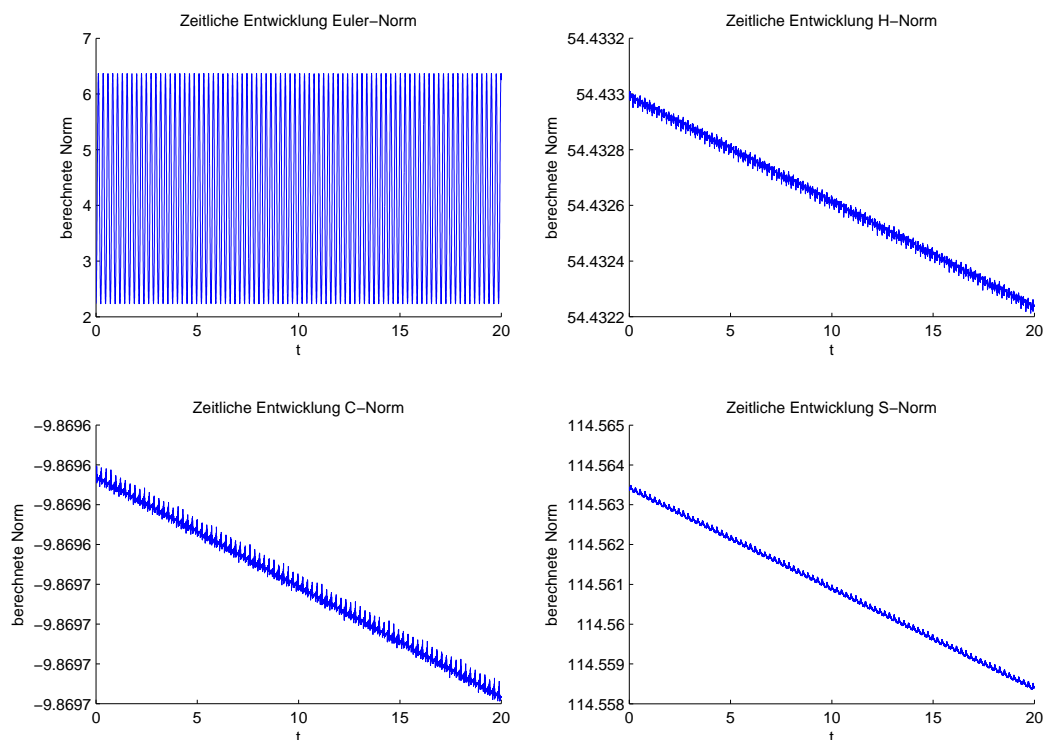


Abbildung 6.11: Zeitliche Entwicklung der Erhaltungsgrößen

Im Gegensatz zur Eulernorm bleiben die drei Erhaltungsgrößen in Abb. 6.11<sup>7</sup> des Systems im Großen und Ganzen konstant. Die erkennbaren Schwankungen und die kontinuierliche Abnahme während der Integration bewegen sich im Bereich von Tausendstel und sind auf die Art der numerischen Integration zurückzuführen. Zwar gäbe es theoretisch Integrationsfunktionen, die auch Erhaltungsgrößen berücksichtigen, allerdings würden Anstrengungen in diese Richtung an den hier durchgeführten Untersuchungen nichts ändern.

Weiters stellt sich die Frage nach der Größe des Analyseschrittes. Wird  $dt\_ana=0.5$  gewählt, wie es beim 3-dimensionalen Modell keinerlei Schwierigkeiten bereitet, so geschieht folgendes:

---

<sup>7</sup>Quelle: Code auf Seite 189

## 6 Ergebnisse

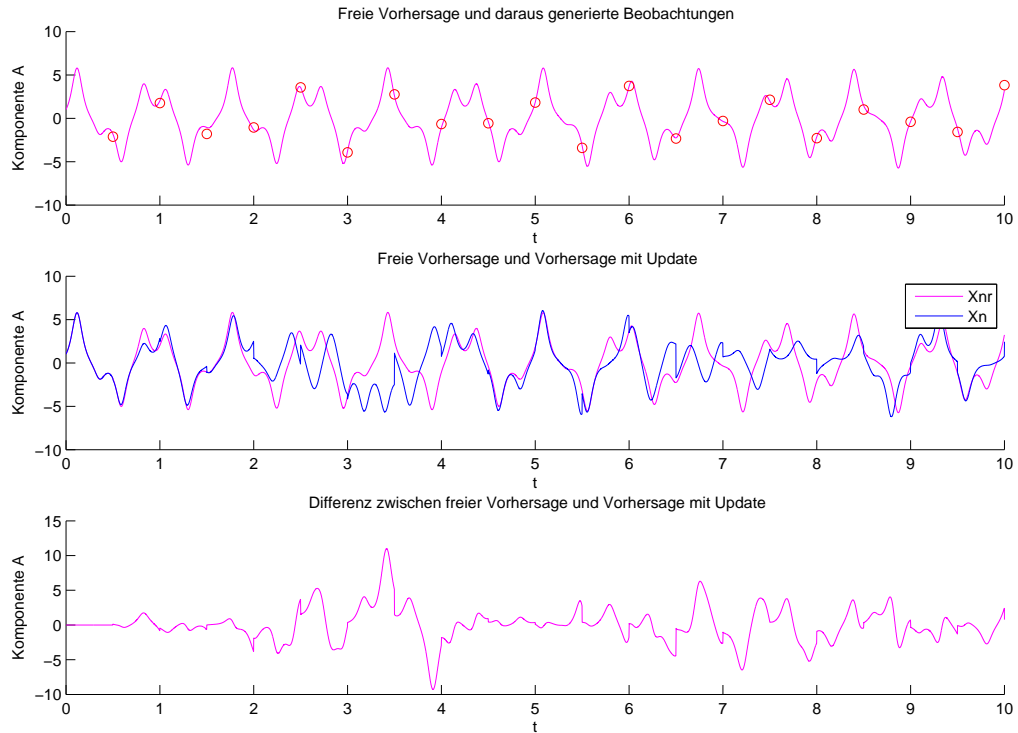


Abbildung 6.12: Freie Vorhersage - Vorhersage mit Updateschritten,  $dt\_ana=0.5$

Alle Beobachtungen werden vor der eigentlichen Berechnung entlang einer freien Vorhersage generiert, dies zeigt die oberste Abbildung in 6.12<sup>8</sup>.

Nun beginnt die eigentliche Systementwicklung, der Anfangszustand wird nichtlinear integriert, zum Analysezeitpunkt wird die zuvor erstellte Beobachtung mittels Kalmanfilter mit dem aktuellen Zustand kombiniert. Nach dem Update-Schritt wurde der Zustand also verändert und wird nun wieder 0.5 Zeiteinheiten weiterentwickelt.

Das System ist jedoch so sensibel, dass sich die Trajektorie bei der Integration dieses leicht veränderten Zustandes über einen halben Zeitschritt sehr weit von der freien Zeitreihe entfernen kann. In der dargestellten Zeitreihe für die erste Komponente geschieht dies zum Beispiel bei etwa  $t = 3.5$ , die beiden Zustände liegen in komplett anderen Regionen des Attraktors, der Plot der Differenzen zeigt einen Unterschied von über 10 Einheiten.

An dieser Stelle zeigt sich nun die höhere Sensibilität des 6-dimensionalen Modells bezüglich Störungen im Anfangszustand gegenüber dem originalen Gleichungssystem in 3 Dimensionen, auf die schon der Vergleich der berechneten Lyapunov-Exponenten deutlich hingewiesen hat.

---

<sup>8</sup>Quelle: Code auf Seite 189

## 6 Ergebnisse

Wird dieses Problem nicht berücksichtigt, so ergibt die Verifikation sehr schlechte Werte, da die Beobachtungen immer wieder sehr weit von der Prognose entfernt sind.

So muss der Analysezeitschritt verkleinert werden und zeigt schließlich bei  $dt\_ana=0.1$  akzeptable Entwicklungen:

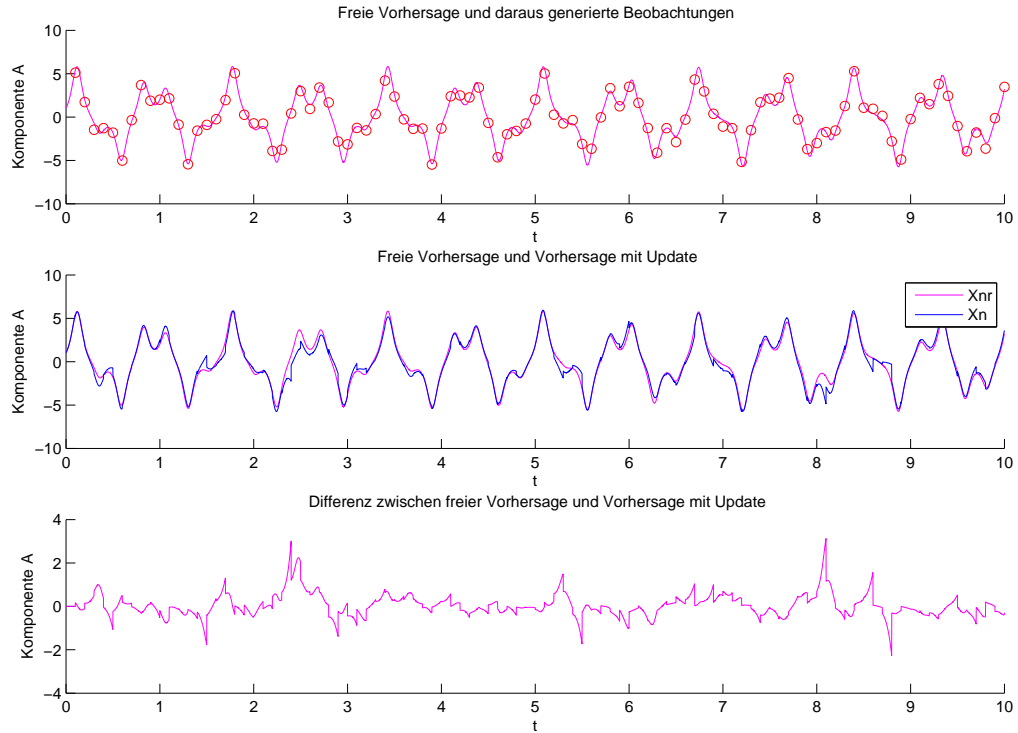


Abbildung 6.13: Freie Vorhersage - Vorhersage mit Updateschritten,  $dt\_ana=0.1$

Wird nun allerdings die neue Parameterkonfiguration benutzt, deren Einführung zur Verwendung der Gewichtungsmatrizen für die Normen notwendig war, so zeigt selbige Untersuchung an dieser Stelle andere Ergebnisse. Da das Modell für diese Parameter eher stabiler ist, sind unter Umständen sogar größere Prognoseintervalle möglich, wieder wird zunächst ein Zeitschritt von 0.5 Einheiten getestet:



## 6 Ergebnisse

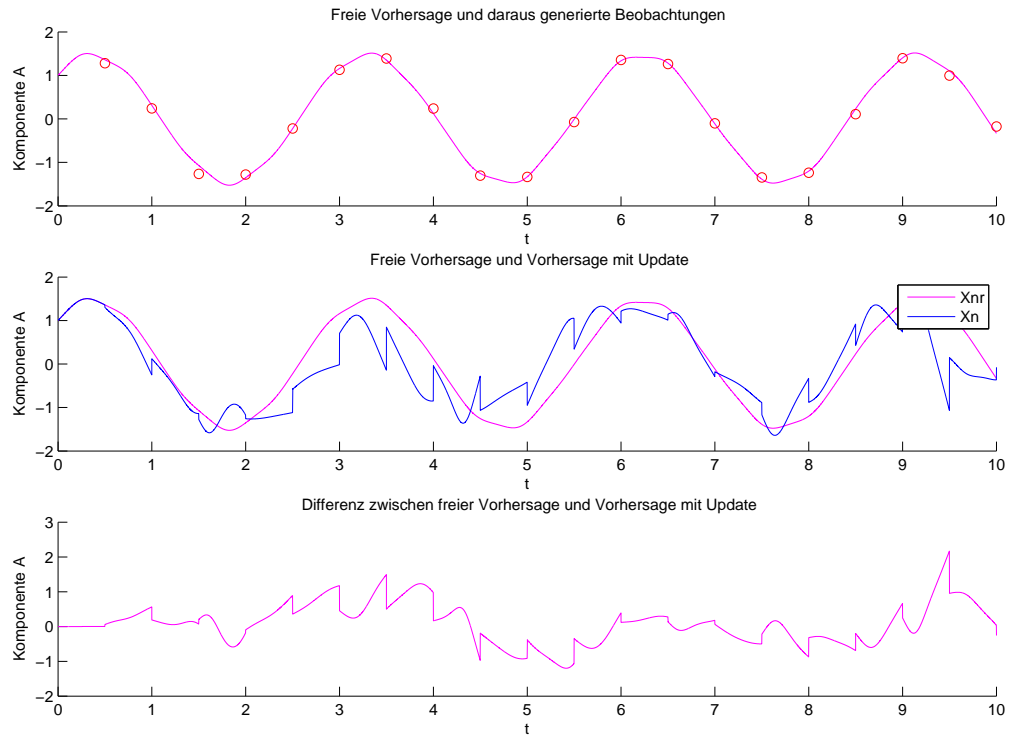


Abbildung 6.14: Freie Vorhersage - Vorhersage mit Updateschritten,  $dt\_ana=0.5$

Unter diesen Bedingungen scheinen tatsächlich größere Zeitschritte möglich zu sein. Es ist in dieser Darstellung im Übrigen auch der im Kapitel 5.2.2.3 erwähnte seltenere Vorzeichenwechsel zu erkennen.

Weiters ist jedoch zu bedenken, dass beim Generieren der Störungen von einem linearen Wachstum ausgegangen wird um die singulären Vektoren der Resolvente verwenden zu können. Darum wird überprüft, ob die angenommene lineare Entwicklung mit der tatsächlich durchgeführten Integration übereinstimmt, wie auf Seite 88 beschrieben.

Für  $dt\_ana=0.5$  (modifizierte Modelleinstellungen) führt dies zu Ergebnissen dieser Größenordnung:

```
ans =
Verhältnis:6.206600
ans =
9500 normB 15.776704
```

Es ist deutlich zu sehen, dass das lineare und das nichtlineare Störungswachstum nicht

übereinstimmen, um dieses Problem zu beheben werden auch im stabileren Modell kleinere Zeitschritte verwendet.

Für  $dt\_ana=0.1$  führt dies zu akzeptablen Ergebnissen wie beispielsweise:

```
ans =
Verhältnis:3.891001
ans =
8200  normB 3.955559
```

Dabei ist noch zu erwähnen, dass die Norm der Resolvente das Wachstum für eine Störung mit Norm 1 liefert, die tatsächlich verwendeten Störvektoren nur Normen im Mittel eher kleiner als 0.5 haben und somit in ihren Wachstumseigenschaften auf jeden Fall hier nun wirklich linear approximiert werden können.

Im Weiteren verhalten sich die beiden Modelleinstellungen wieder gleich, es werden wieder die Standardeinstellungen für Darstellungen verwendet.

Auch die Stabilität der Fehlerkovarianzmatrix  $P$  sollte überprüft werden, steigt diese plötzlich um viele Größenordnungen an, so verliert das System gleichzeitig jegliches Vertrauen in seine eigene Prognose. Solche Instabilitäten können auf Implementierungsfehler im Zusammenhang mit der Resolvente  $A$ , der einzelnen linearen Propagatoren  $M$  oder noch weiter zurückführend, der verwendeten Jacobi-Funktion hinweisen. Ist die Fehlerkovarianzmatrix zu groß weil die Analyseintervalle zu weit auseinander liegen, so wird zwar die Prognose korrekt berechnet, geht dann aber in den Update-Schritt der Analyse fast nicht ein.

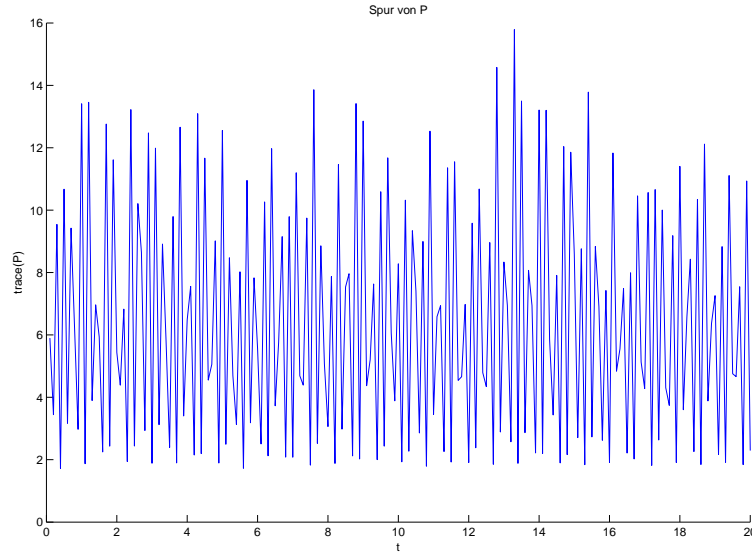


Abbildung 6.15: Spur der Fehlerkovarianzmatrix  $P$  vor dem Update

Die Spur von  $P$  (siehe Seite 70) in Abb. 6.15<sup>9</sup> zeigt zwar immer wieder unsicherere Stellen im Modell, bewegt sich aber in einem durchaus akzeptablen Rahmen.

## 6.6 Größe der Störungen

Auch die Größen der Störungen spielen eine wichtige Rolle im EPS, sie sollen so gewählt werden, dass der Ensemblespread am Ende der Integration in etwa die Modellunsicherheit repräsentiert.

Zum Ersten geht die momentane Sensibilität der Vorhersage durch die Multiplikation der Zufallsmatrix zur Störungsskalierung mit der Wurzel der Spur von  $P$  ein. Befinden sich die Zustandstrajektorien an einem empfindlichen Ort des Attraktors, so ist die Spur der Fehlermatrix  $P$  groß, somit auch deren Wurzel und auch die generierten Störungen, was schließlich zu dem erwünschten größeren Ensemblespread für schwierig vorherzusagende Systemzustände führt.

Außerdem kann der Ensemblespread direkt mit dem jeweiligen rms-Wert verglichen werden und bei zu großen Unterschieden durch einen empirischen Faktor angepasst werden.

Abb. 6.16<sup>10</sup> zeigt sowohl die Spur von  $P$  vor dem Update-Schritt als auch die spread- und rms-Werte entlang einer verkürzten Zeitreihe:

<sup>9</sup>Quelle: Code auf Seite 189

<sup>10</sup>Quelle: Code auf Seite 189

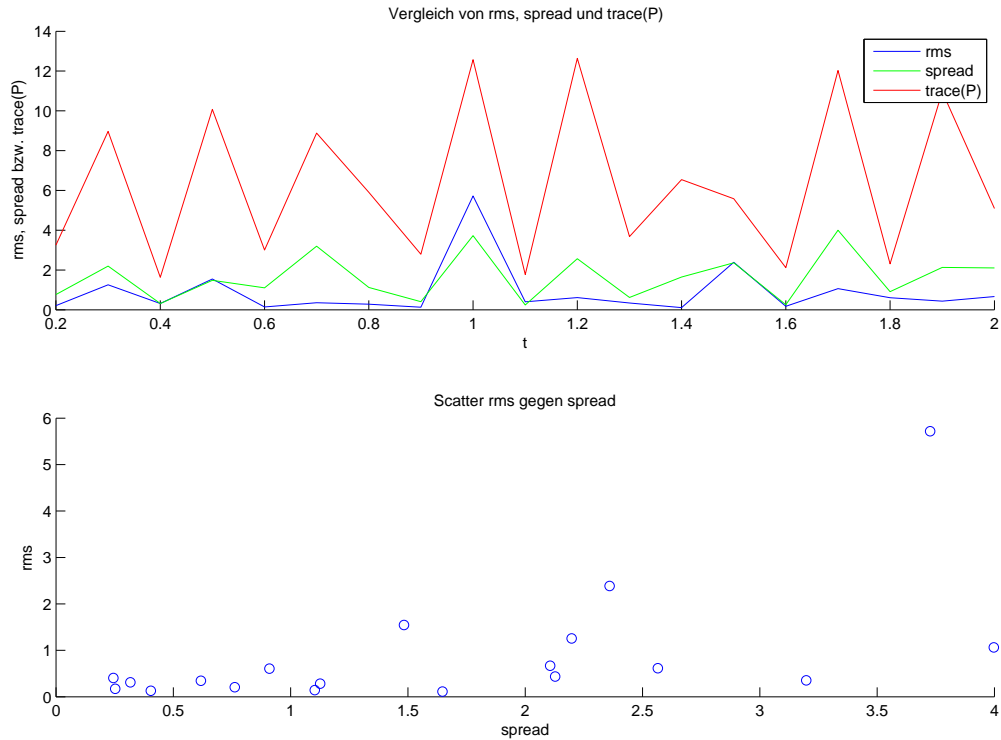


Abbildung 6.16: Korrelation rms, spread und Spur(P)

Die Korrelation dieser drei Größen ist deutlich sichtbar, das EPS kann also Prognoseunsicherheiten identifizieren und mit angemessenen Störungen reagieren. Auch die Größenordnung von spread und rms stimmt im Mittel sehr gut überein, der empirische Faktor ist nicht notwendig bzw. kann auf eins gesetzt werden.

## 6.7 Bewertung des EPS - Optimierung durch den Energie-Vorticity-Casimir

### 6.7.1 Brier Score

Das vorrangige Ziel dieser Arbeit ist es, die Vorteile eines Energie-Vorticity-Casimirs (HC-Norm) gegenüber einem reinen Energie-Casimir (H-Norm) zur Generierung der Ensemblestörungen zu zeigen.

Alle folgenden Abbildungen stammen aus Berechnungen mit den Codes auf Seite 189 und auf Seite 229.

Dazu wurde das Programm `rbc_6dim.m` mit folgender Parameterkonfiguration gestartet:

```
dt=0.001;
t_end=20;
dt_ana=0.1;
n=30;
```

Natürlich wurden ab hier nur noch die eigens dafür ermittelten angepassten Parameter-einstellungen verwendet:

```
Xn=[1 1 4 1 1 1]
b=1.102
```

Das Hauptergebnis ist in Abb. 6.17 anhand eines Brier Score-Vergleichs der beiden Normen dargestellt. Die Verifikation wurde zwar komponentenweise durchgeführt, zu aller erst soll jedoch das Mittel<sup>11</sup> des gesamten Brier-Scores über alle Komponenten gezeigt werden. Getestet wurden zunächst alle  $c1$  von  $-5$  bis  $+5$  in Abständen 1. Eine Optimierung liegt immer dort vor, wo der  $BS$  der HC-Norm kleiner ist als jener der H-Norm.

Wichtig sind nicht die genauen Werte der  $c1$ , sondern die Tatsache, dass die gewünschten Verbesserungen theoretisch möglich sind und in der angewandten Vorhersage vorteilhaft eingesetzt werden könnten.

---

<sup>11</sup>mit "Mittel" ist eine gleich gewichtete Norm gemeint

## 6 Ergebnisse

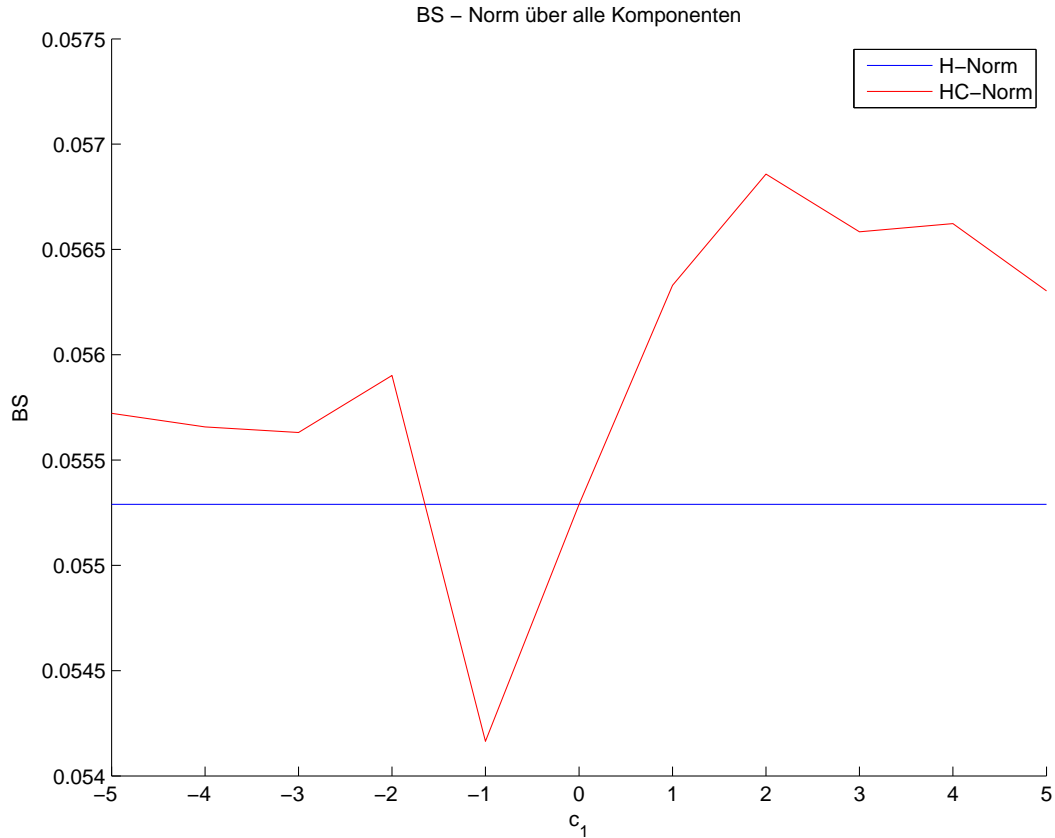


Abbildung 6.17: BS der gesamten Prognose,  $c_1 = [-5, 5]$

Der Brier Score (siehe Gl. (5.29)) der kombinierten Energie-Vorticity-Norm zeigt für  $c_1 = -1$  eine signifikante Verbesserung gegenüber der reinen Energienorm. Damit ist die aufgestellte These der Arbeit eindeutig bewiesen.

Der detaillierte Programm-Output zu diesem Wert kann im Anhang auf Seite 155 nachgelesen werden.

Besonders positiv ist auch die unimodale Verteilung dieser Optimierung zu bemerken: Erinnert sich der Leser an die geometrische Interpretation der Erhaltungsgrößen, so soll behauptet werden, dass der Koeffizient  $c_1$  der Normkonstruktion die „relative Geometrie“ der „Erhaltungsflächen“ erfassen soll.

Dies ist zwar keine bewiesene Tatsache, liegt aber aus folgendem Grund nahe:

Das Modell verfügt über drei Erhaltungsgrößen, diese müssen in der Systemdynamik theoretisch alle gleichberechtigt sein. Wäre deren Relation zueinander für die Generierung der EPS-Störungen nicht von Bedeutung, so müsste die gleichgewichtete Linearkombination aller Erhaltungsgrößen optimale Ergebnisse erzielen. In Kapitel 6.8.2 wird dies jedoch eindeutig widerlegt. Dass die Berücksichtigung einer bestimmten Erhaltungsgröße eines deterministischen Gleichungssystems global wichtiger ist als die einer anderen ist unwahrscheinlich, darum sei an dieser Stelle die Hypothese aufgestellt, dass die zu-

standsabhängige Relation der Erhaltungsgrößen zueinander ausschlaggebend ist. Mit „relativer Geometrie“ sei also nun für den 3-dimensionalen Fall der Schnittwinkel angenommen, das 6-dimensionale Pendant zu einem solchen bleibt im Moment der Phantasie überlassen. Da sich die Flächen der konservativen Funktionen jedoch nicht überall am Attraktor in gleicher geometrischer Beziehung begegnen, kann ein konstant approximiertes  $c_1$  nur ein *first guess* für die statistisch am häufigsten vorkommende Relation sein. Gäbe es zum Beispiel für den Erfolg von  $c_1$  zwei verschiedene bevorzugte Werte, so wäre der Vorteil jedes einzelnen nur etwa halb so groß, eine Mittelung der beiden wäre absolut sinnlos.

So erzielt die unimodale  $c_1$ -Erfolgsverteilung für den Wert  $c_1 = -1$  im Rahmen dieses Konzepts optimale Verbesserungen.

Weiters soll die Aufspaltung des  $BS$  in Zuverlässigkeit und Auflösung betrachtet werden:

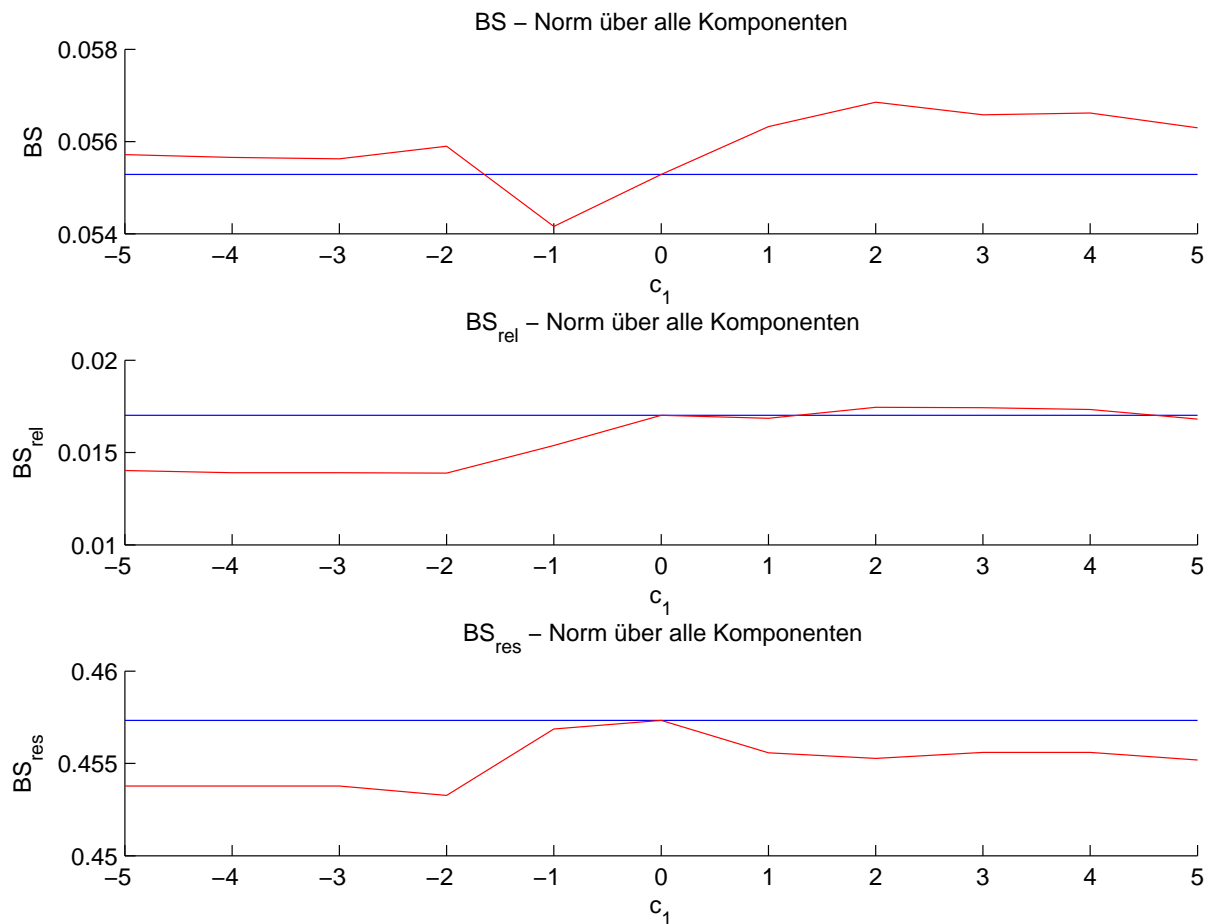


Abbildung 6.18:  $BS$  der gesamten Prognose, Aufspaltung in Zuverlässigkeit und Auflösung,  $c_1 = [-5, 5]$

Die gesamte gezeigte Verbesserung des  $BS$  stammt aus der Zuverlässigkeit, die Auflösung wird für das gewählte  $c_1 = -1$  jedoch nur minimal schlechter.

## 6 Ergebnisse

Ja nach Anforderungen an das Modell und an die prognostizierten Parameter kann  $c_1 = -2$  die Zuverlässigkeit noch weiter steigern, wenn dafür eine Verschlechterung der Auflösung in Kauf genommen wird.

Da das Modell 6-dimensional ist, sollen auch die Einflüsse auf die einzelnen Komponenten näher betrachtet werden, jeder der drei obigen Plots wird nun in die dynamischen Variablen  $A$  bis  $F$  aufgespalten.

Eine gleichzeitige Verbesserung aller Komponenten kann trotz Gesamterfolg hier nicht gezeigt werden und gilt als weiteres Indiz für die komplexe nichtlineare Modelldynamik:

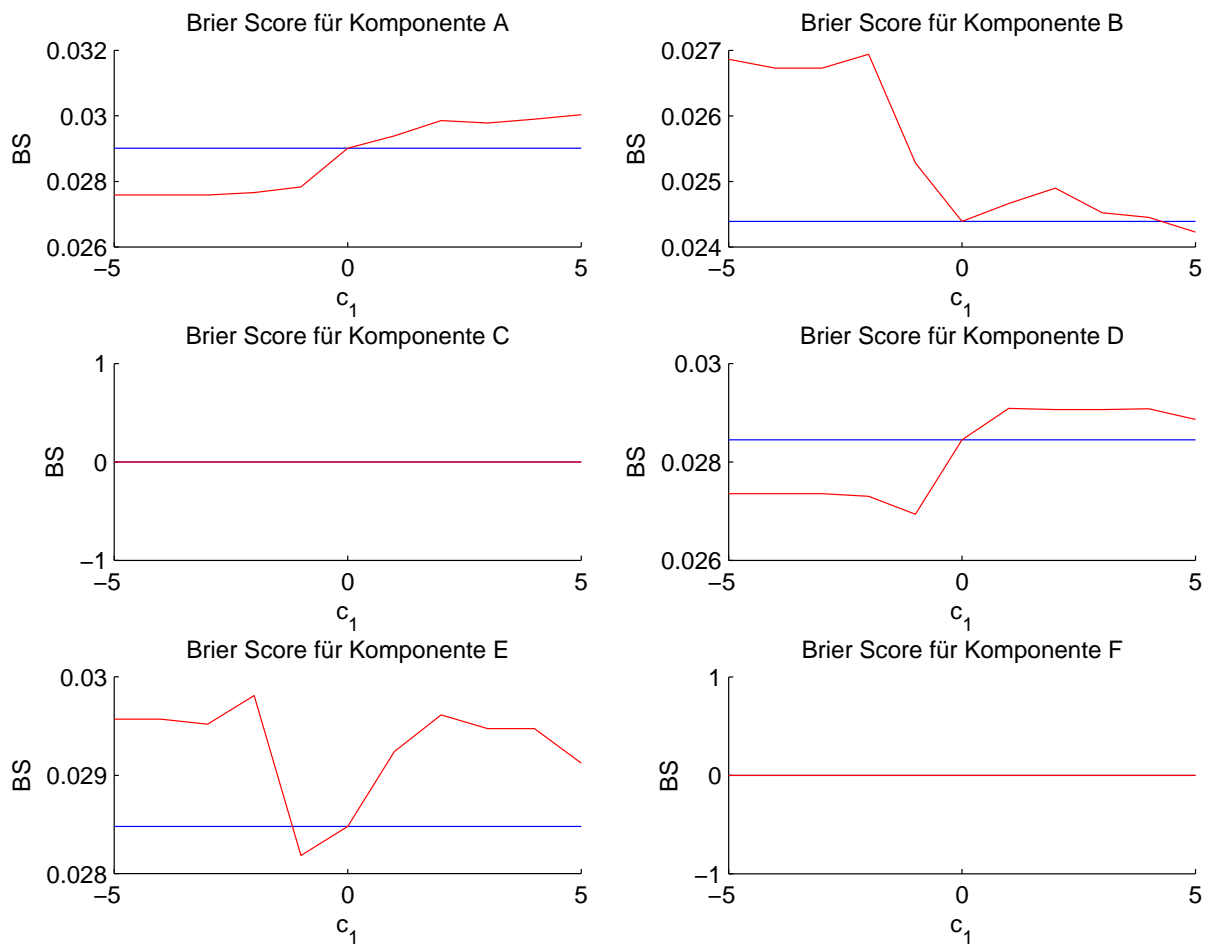


Abbildung 6.19: BS komponentenweise,  $c_1 = [-5, 5]$



## 6 Ergebnisse

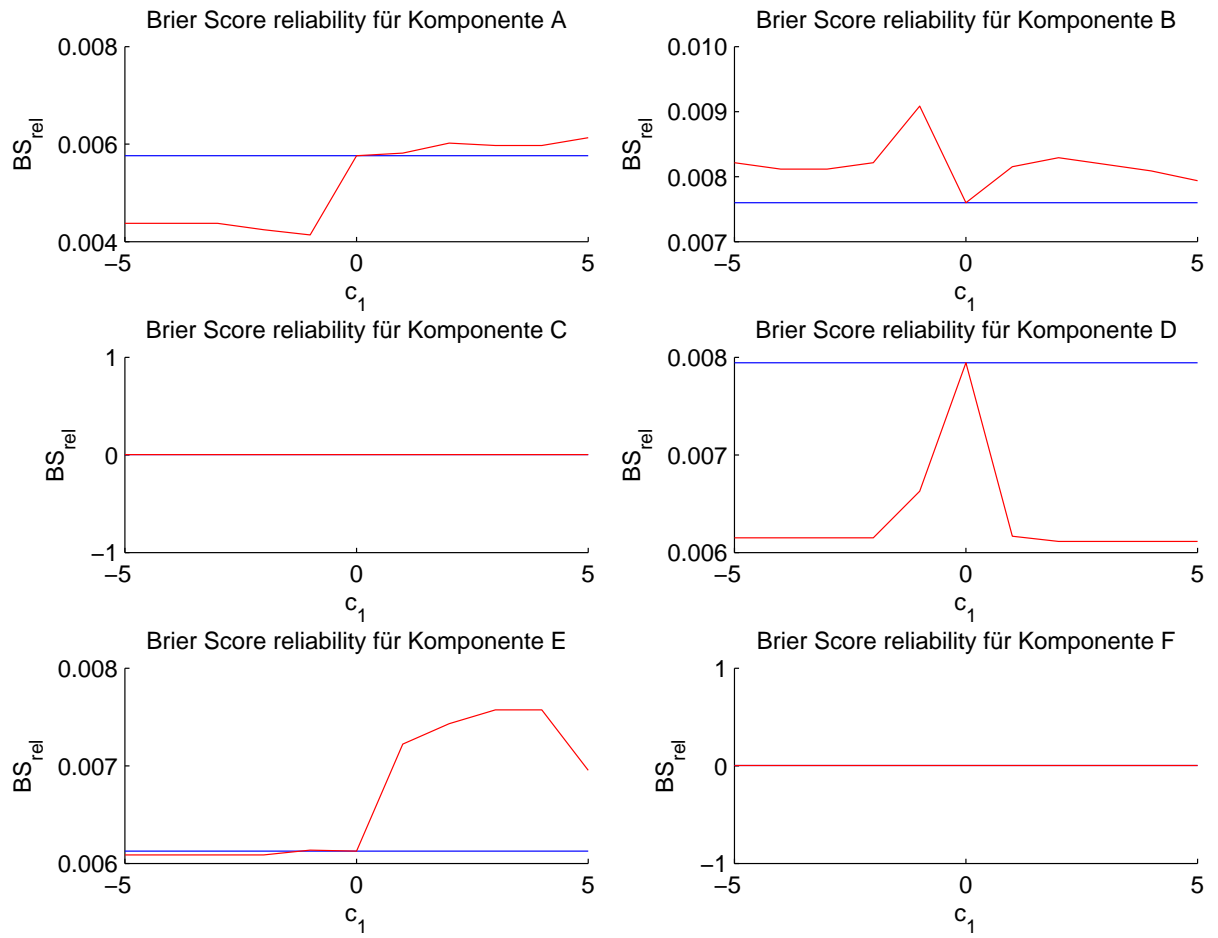


Abbildung 6.20: BS reliability, komponentenweise,  $c_1 = [-5, 5]$

## 6 Ergebnisse

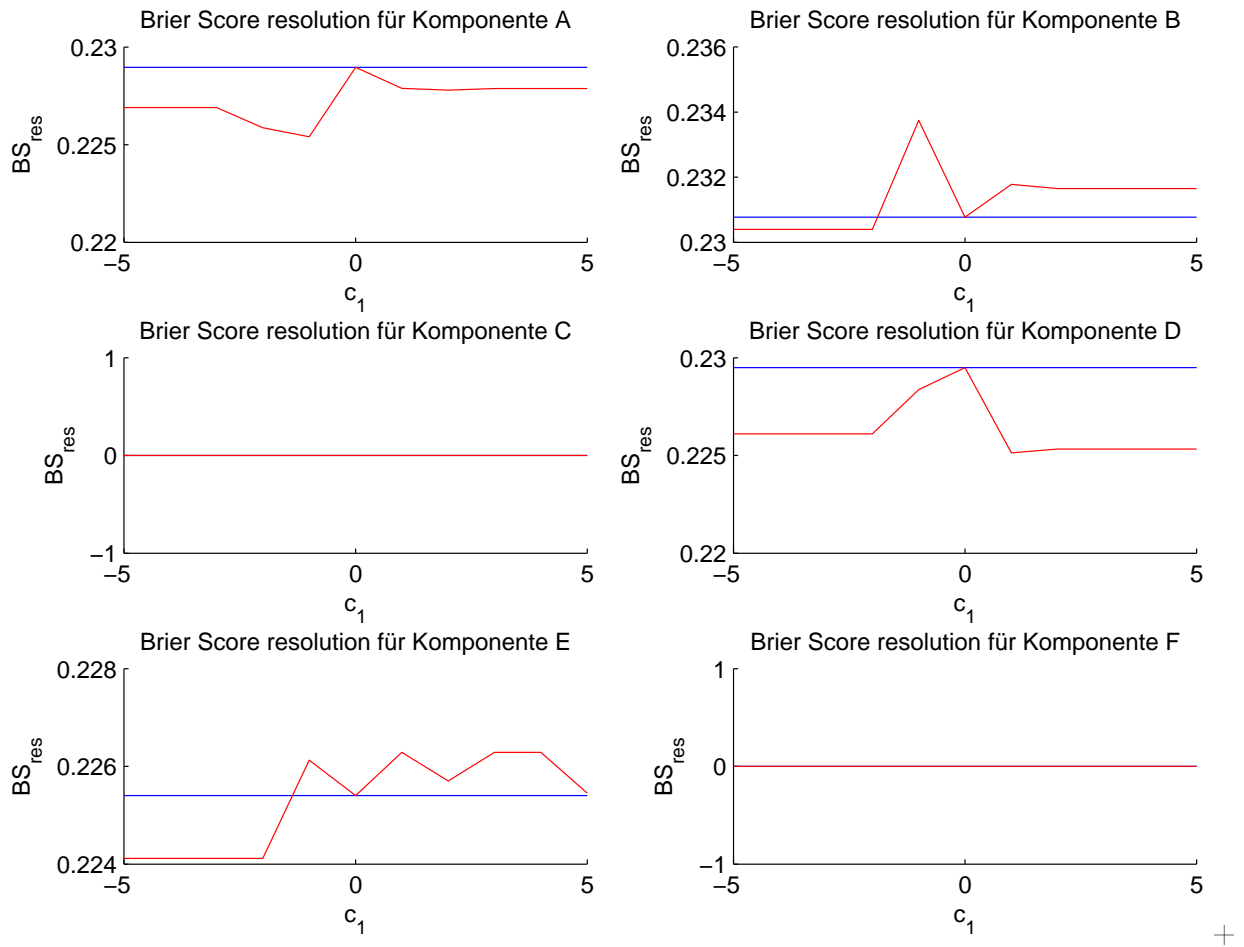


Abbildung 6.21: BS resolution, komponentenweise,  $c_1 \in [-5, 5]$

Eine zusammengefasste Aussage der gezeigten Grafiken wird in Abb. 6.22 schemenhaft dargestellt, wobei grün eine Verbesserung und rot eine Verschlechterung gegenüber der reinen Energie-Norm anzeigt:

**c1=-1**

	BS	BS_rel	BS_res
A	grün	grün	rot
B	rot	rot	grün
C	-	-	-
D	grün	grün	rot
E	grün	rot	grün
F	-	-	-

Abbildung 6.22: Brier Scores für  $c=-1$  im Vergleich zur Energienorm, komponentenweise

## 6 Ergebnisse

Es scheint, als würde die erweiterte Norm in manchen Fällen die Auflösung der Modellvorhersage auf Kosten der Zuverlässigkeit verbessern und umgekehrt. Interessant ist, dass genau die physikalisch zusammengehörigen Größen sich reliability und resolution betreffend konträr verhalten.

Trotz einiger roter Flächen in Abb. 6.22 sei an die insgesamt Optimierung erinnert, eine Darstellung der Norm über alle Komponenten soll die quantitative Dominanz der Verbesserungen gegenüber den Verschlechterungen im gesamten Brier Score und der Zuverlässigkeit zeigen.

c1=-1			
	BS	BS_rel	BS_res
Norm über alle Komp.	-	-	-

Abbildung 6.23: Brier Scores für  $c1=-1$ , Norm über alle Komponenten im Vergleich zur Energienorm

Dieses Ergebnis kann allerdings noch durch eine genauere Bestimmung des optimalen Wertes für  $c1$  verbessert werden.

So liegt es nahe das Programm ein weiteres Mal laufen zu lassen und diesmal alle  $c1$  zwischen  $-2$  und  $0$  (in,  $0.1$  Schritten) auszutesten. Die folgenden Plots zoomen sozusagen in die vorherigen Reihen der  $BS$  hinein und zeigen, was zwischen diesen beiden Werten geschieht:

## 6 Ergebnisse

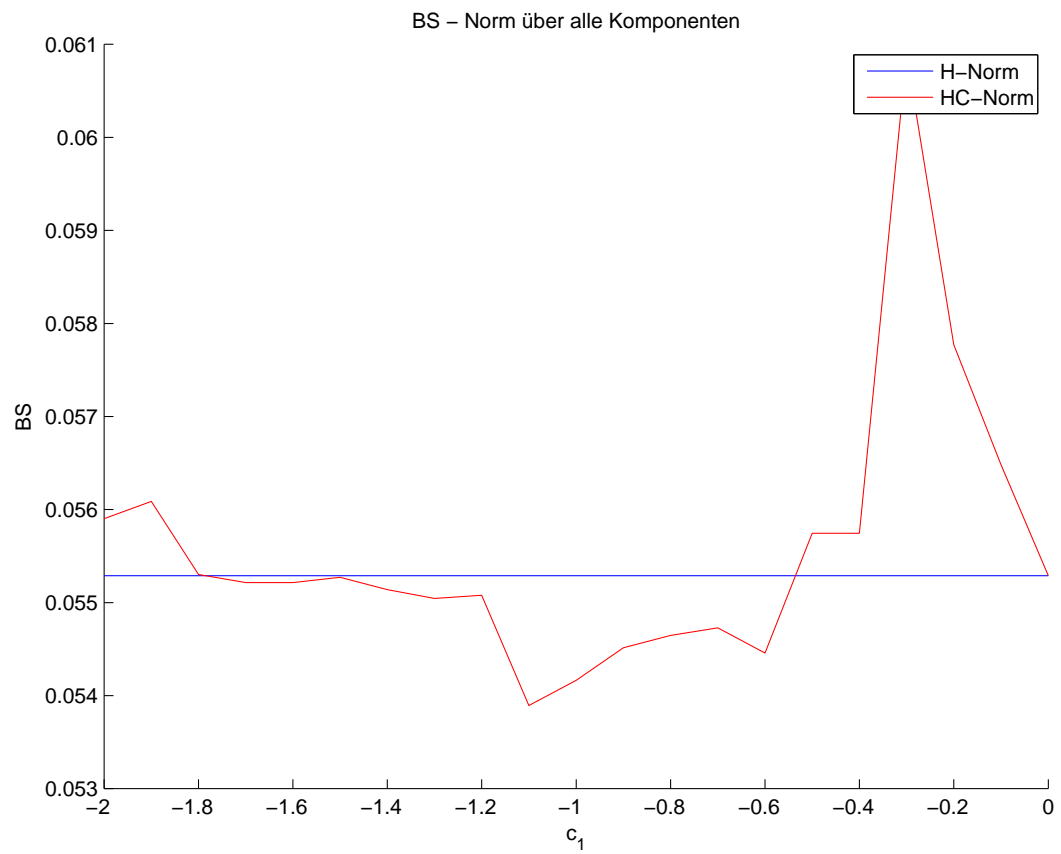


Abbildung 6.24: BS der gesamten Prognose,  $c_1 = [-2, 0]$

## 6 Ergebnisse

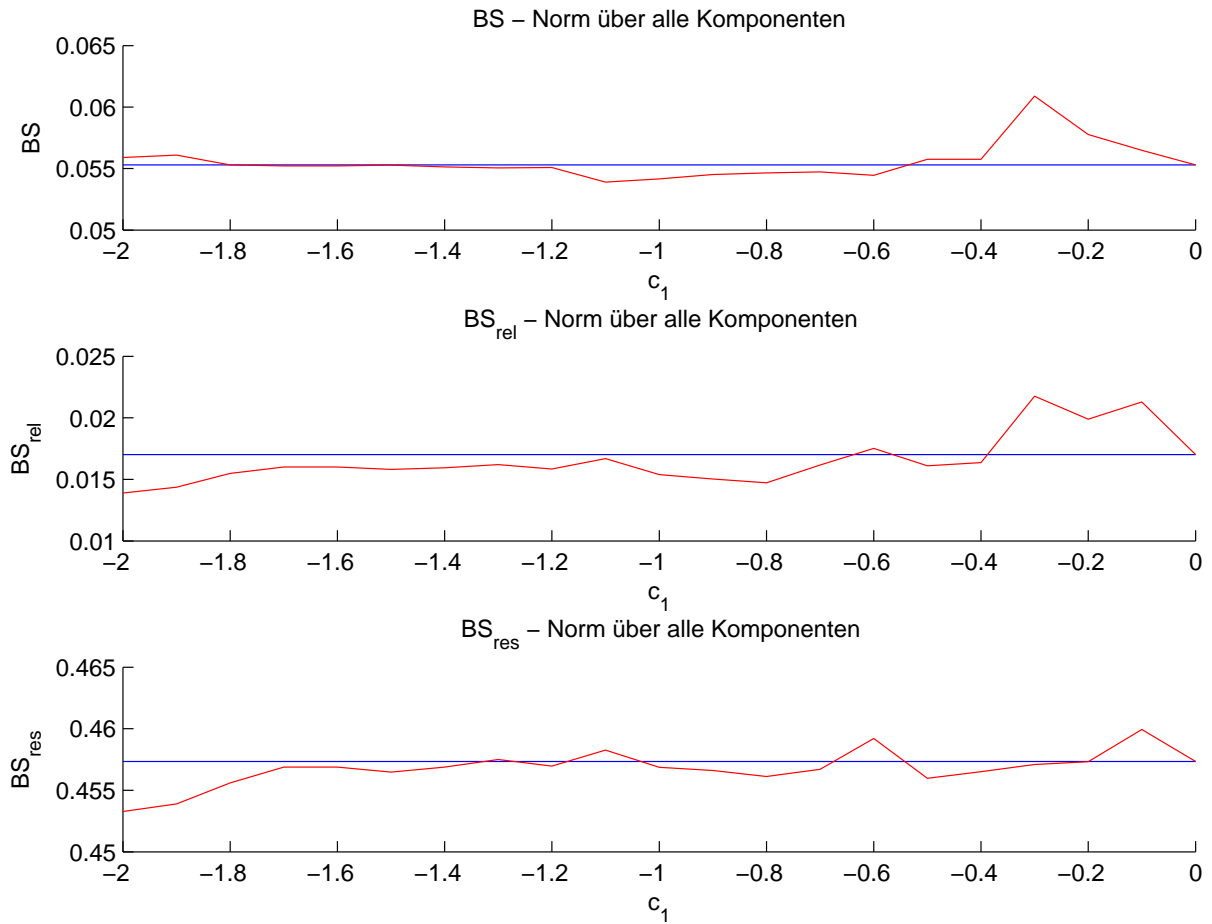


Abbildung 6.25: BS der gesamten Prognose, Aufspaltung in Zuverlässigkeit und Auflösung,  $c_1 = [-2, 0]$

Bereits diese Darstellung liefert erfreuliche Ergebnisse, für  $c_1 = -1.1$  wird sowohl Zuverlässigkeit als auch Auflösung der Vorhersage verbessert, was vorherige Berechnungen nicht auflösen konnten. (siehe auch 158)

Weiters wieder die Betrachtung der einzelnen Komponenten:

## 6 Ergebnisse

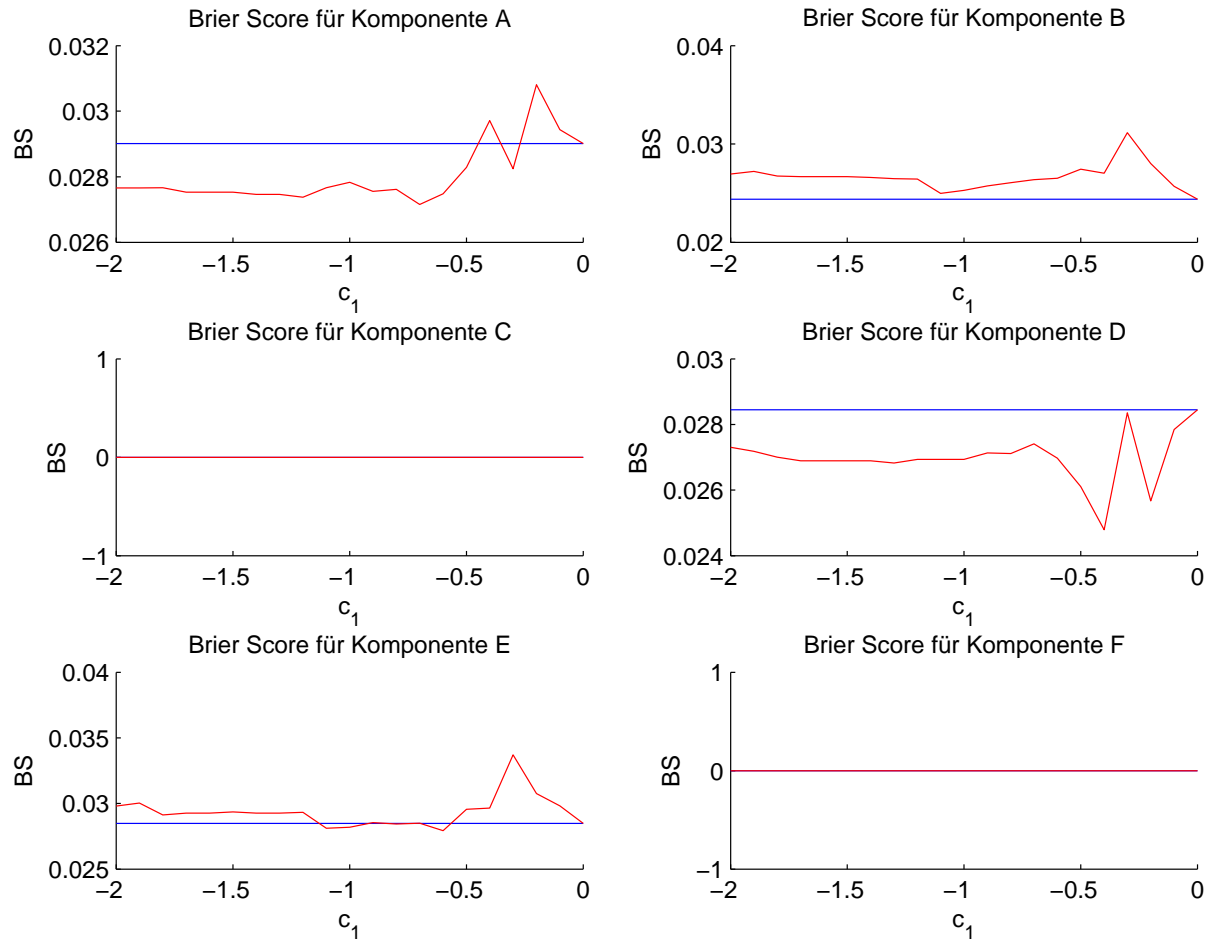


Abbildung 6.26: BS komponentenweise,  $c_1 = [-2, 0]$

## 6 Ergebnisse

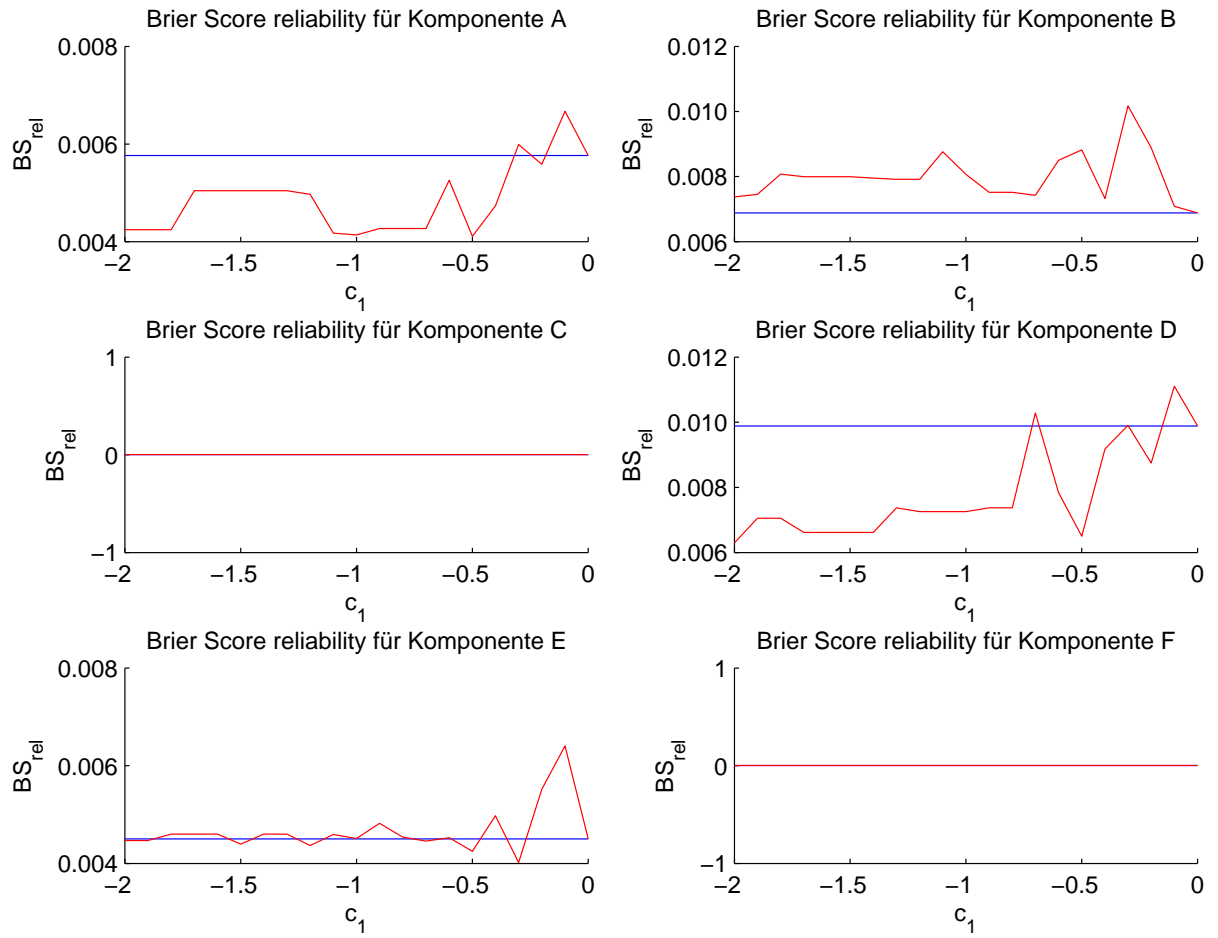


Abbildung 6.27: BS reliability, komponentenweise,  $c_1 = [-2, 0]$

## 6 Ergebnisse

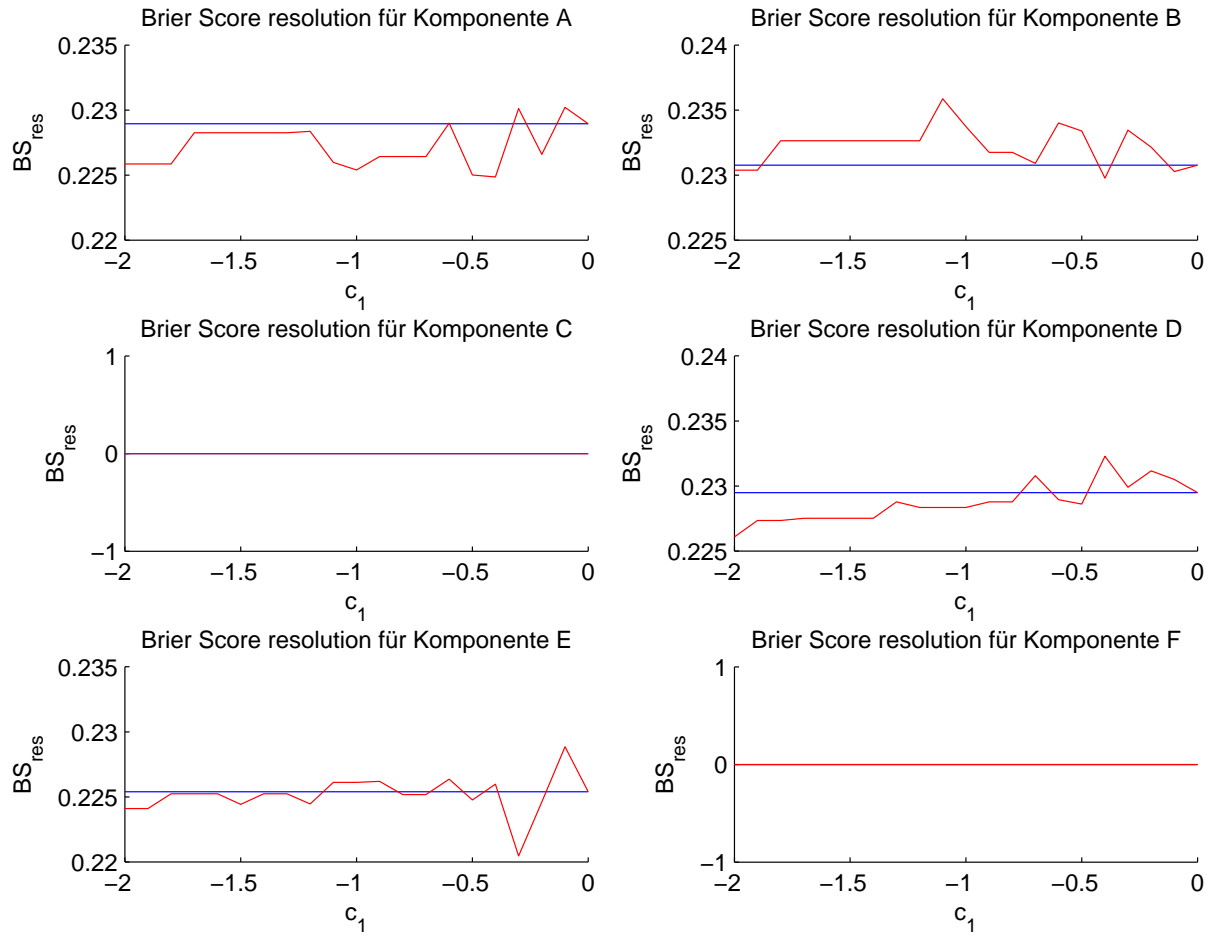


Abbildung 6.28: BS resolution, komponentenweise,  $c_1 = [-2, 0]$

Der schemenhafte Vergleich der Scores Abb. 6.22 wird nun für  $c_1 = -1.1$  in 6.29 wiederholt, zeigt allerdings den erzielten Erfolg noch nicht:

**c1=-1.1**

	BS	BS_rel	BS_res
A			
B			
C	-	-	-
D			
E			
F	-	-	-

Abbildung 6.29: Brier Scores für  $c = -1.1$  im Vergleich zur Energienorm, komponentenweise



Nun soll jedoch noch eine physikalisch relevantere Interpretation, im Gegensatz zur rein mathematischen Auswertung der Ergebnisse, besprochen werden.

Die prozentuell erzielten Verbesserungen bzw. auch teilweise Verschlechterungen der einzelnen Komponenten werden so gewichtet, wie die Fourierkoeffizienten in der Konstruktion der Feldstrukturen selbst (siehe (4.25) und (4.26) ). Dies geschieht im Allgemeinen über die geometrischen Faktoren  $a - f$ , da jeweils  $A$  und  $B$  bzw.  $D$  und  $E$  die gleichen Vorfaktoren haben, muss in diesem Fall nicht extra gewichtet werden.

Für den gesamten Brier Score des Strömungsfeldes bedeutet dies eine insgesamt Verbesserung, da die Verbesserung der Komponente  $A$  die Verschlechterung der Komponente  $B$  deutlich übersteigt.

Die Ergebnisse dieser Analyse sind in Abb. 6.30 schematisch dargestellt. Lediglich die Auflösung des Temperaturfeldes zeigt keine eindeutige Verbesserung, Komponente  $D$  und  $E$  werden durch  $d$  gleich gewichtet, der negative Einfluss von  $D$  übersteigt jedoch den positiven von  $E$ .

**c1=-1.1**

	BS	BS_rel	BS_res
Strömungs- Feld			
Temperatur- Feld			

Abbildung 6.30: Verbesserungen der physikalischen Feldprognosen für  $c1=-1.1$  im Vergleich zur Energienorm

Die hellrosarote Farbe im Feld der Auflösung des Temperaturfeldes soll nicht nur Toner sparen, sondern auf eine prozentuell viel geringere Verschlechterung (ca. 0.17%) gegenüber der gesamten Verbesserung von ca. 3.44% hinweisen.

Als Abschluss dieser Analysen wird nun zur Zusammenfassung auch noch die zuvor kurz erwähnte gefundene Entdeckung für  $c1 = -1.1$  dargestellt:

**c1=-1.1**

	BS	BS_rel	BS_res
Norm über alle Komp.	-	-	-

Abbildung 6.31: Brier Scores für  $c1=-1.1$ , Norm über alle Komponenten im Vergleich zur Energienorm

Noch einmal sei gesagt, dass nicht der ermittelte Zahlenwert  $c1=-1.1$  wichtig ist, sondern nur der allgemeine Beweis der Optimierungsmöglichkeiten durch dieses Konzept.

### 6.7.2 Reliability-Diagramm, Resolution-Diagramm

Die grafische Darstellung der Zuverlässigkeit und Auflösung dient weniger dem Vergleich der beiden Normen als der Bewertung des implementierten Modells insgesamt, trotzdem werden beide Ergebnisse eingetragen.

Das Modell wurde zu Gunsten der grafischen Darstellungen etwas länger integriert ( $t_{\text{end}}=100$ ), nachdem der Einfluss des Zufalls jedoch schon zuvor eliminiert wurde, stört das die Vergleichbarkeit der Auswertungen keineswegs.

Im Reliability-Diagramm dargestellt werden die bedingten Wahrscheinlichkeiten für das Eintreten eines Events. Dafür werden die von den Ensemblemembers vorhergesagten Wahrscheinlichkeiten in Klassen geteilt und auf der  $x$ -Achse aufgetragen. Auf der  $y$ -Achse befindet sich dann die tatsächlich beobachtete Wahrscheinlichkeit, bei gegebener Vorhersageklasse. Wenn das EPS das Event mit beispielsweise 80% vorhersagt, dann sollte es auch in 80% all dieser Vorhersagen eintreten, Ziel sind also Eintragungen im Diagramm möglichst nahe der 45° Mediane.

Insgesamt fällt zwar auf, dass sich einige Datenpunkte von der Mediane weit entfernt befinden, allerdings darf dieses Diagramm nie ohne Betrachtung der zugehörigen Auflösung interpretiert werden. In Abb. 6.33 ist zu sehen, dass der Großteil der vorhergesagten Wahrscheinlichkeiten in die Klassen 0 - 10% und 90-100% fallen. Das ist nicht nur sehr gut für die Auflösung selbst, in allen 5 Komponenten liegt der Wert der Zuverlässigkeit für diese beiden Klassen fast optimal. Dabei ist auch noch zu bemerken, dass als

## 6 Ergebnisse

x-Koordinate hier stets die Klassenmitte verwendet wird, also z.B. 5% für die Klasse 0-10%. Da die meisten Vorhersagen aber tatsächlich mit einer Wahrscheinlichkeit von 0 und 100% prognostizieren, liegen diese Punkte eigentlich noch besser als hier dargestellt. Die Punkte weit entfernt von der Mediane fallen nicht besonders ins Gewicht, da sie nur wenige Vorhersagen betreffen.

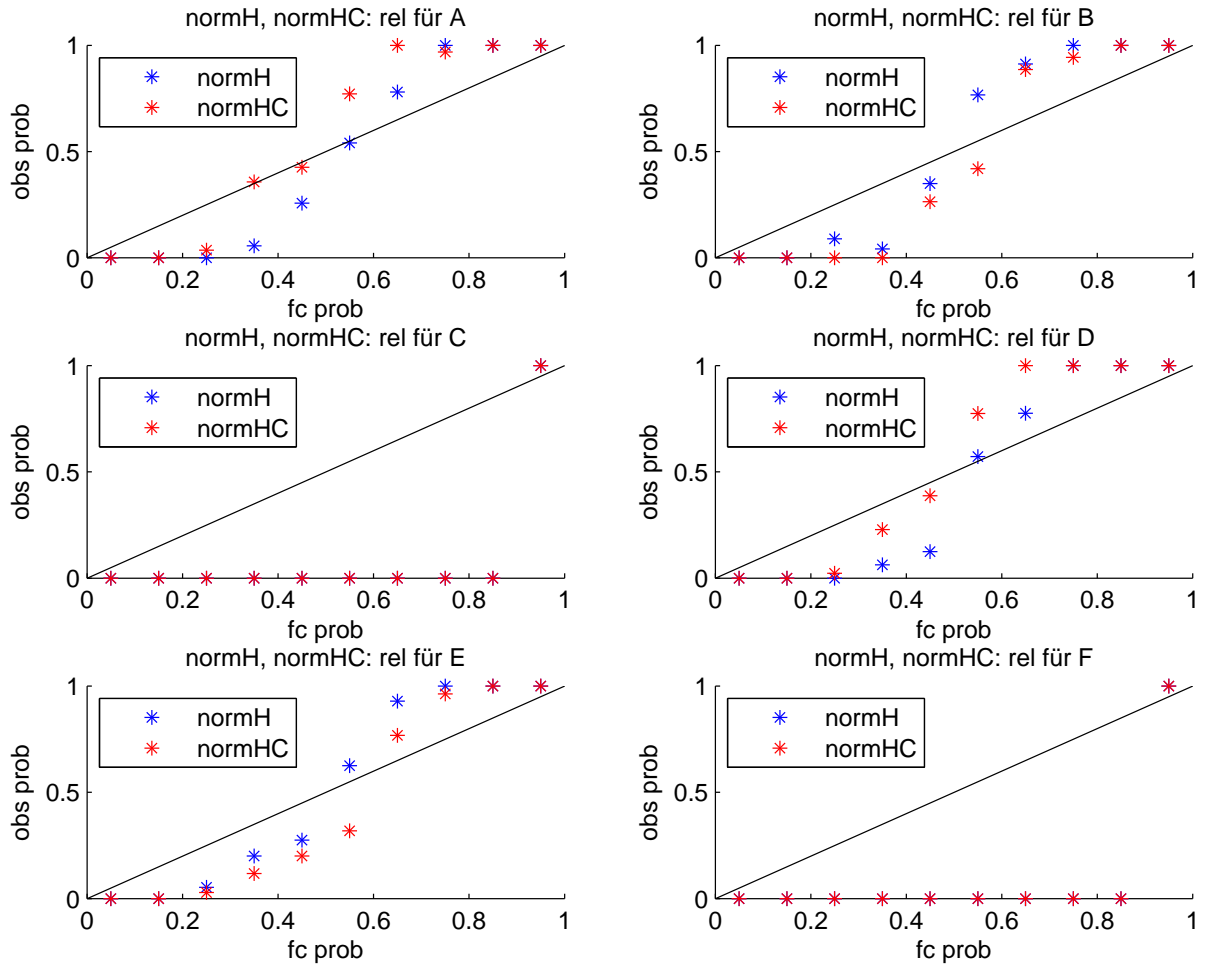


Abbildung 6.32: Reliability-Diagramm

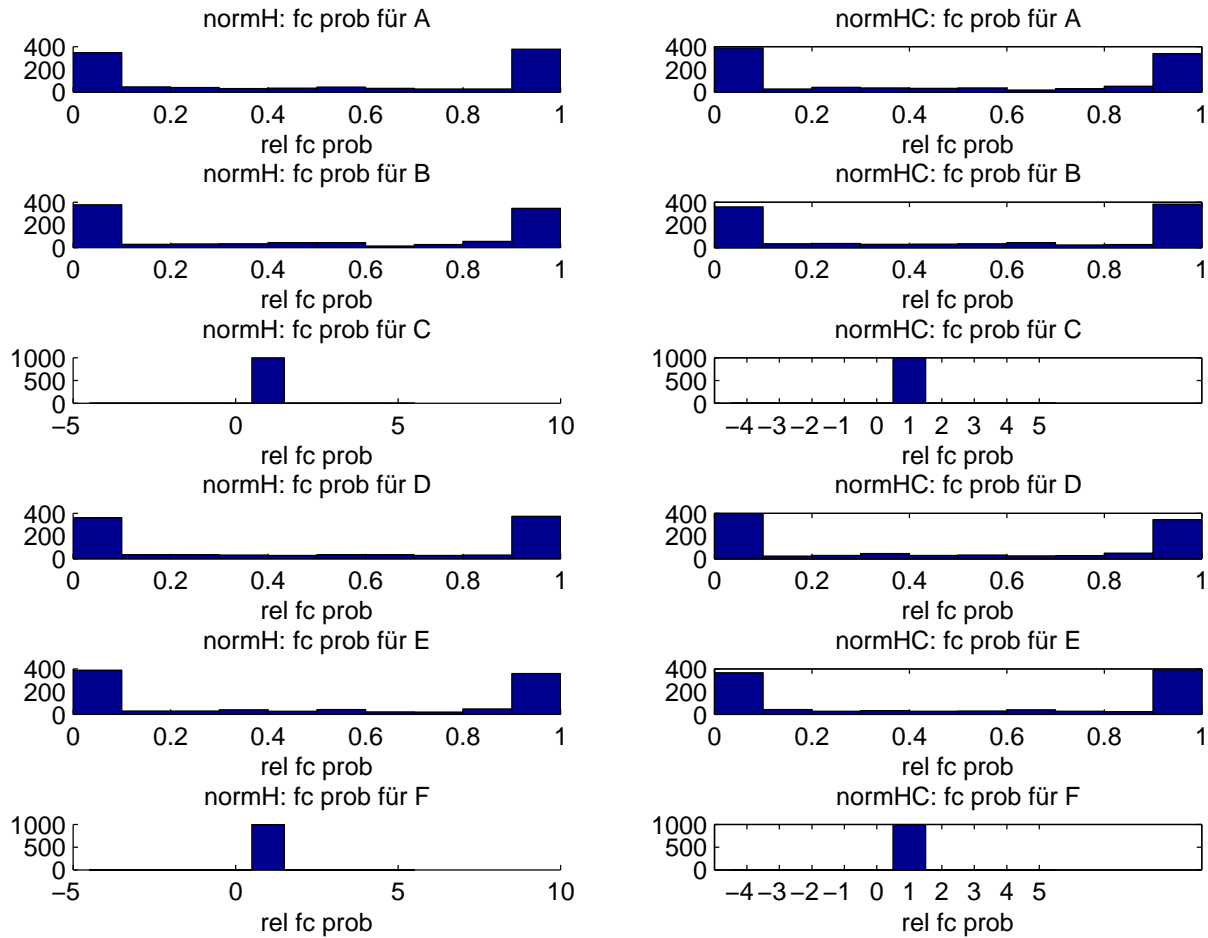


Abbildung 6.33: Histogramm der vorhergesagten Wahrscheinlichkeitsklassen

Der Vollständigkeit halber wurden die Werte der H-Norm und der HC-Norm in dasselbe Diagramm eingetragen, in den meisten Fällen sind die in Kap. 6.7.1 gezeigten Verbesserungen zahlenmäßig jedoch nicht groß genug um grafisch deutlich sichtbar zu sein. Tatsächlich ist es eher nicht ratsam hier einen Vergleich zu ziehen, da für jeden Punkt im Zuverlässigkeitsdiagramm die zugehörige Auflösung betrachtet werden müsste, die jedoch aus der groben Histogramm-Darstellung grafisch nicht sichtbar ist.

### 6.7.3 ROC-Diagramm

Im ROC-Diagramm werden Punkte möglichst nahe dem linken oberen Eck angestrebt, da hier die Erkennungswahrscheinlichkeit 100% beträgt ohne dabei falsche Alarme auszulösen.

Das Modell zeigt in allen Komponenten sehr gute POD-POFD-Werte, ob jetzt die reine Energie-Norm oder die kombinierte Norm bei dieser Bewertung besser abschneidet ist nicht eindeutig zu sehen.

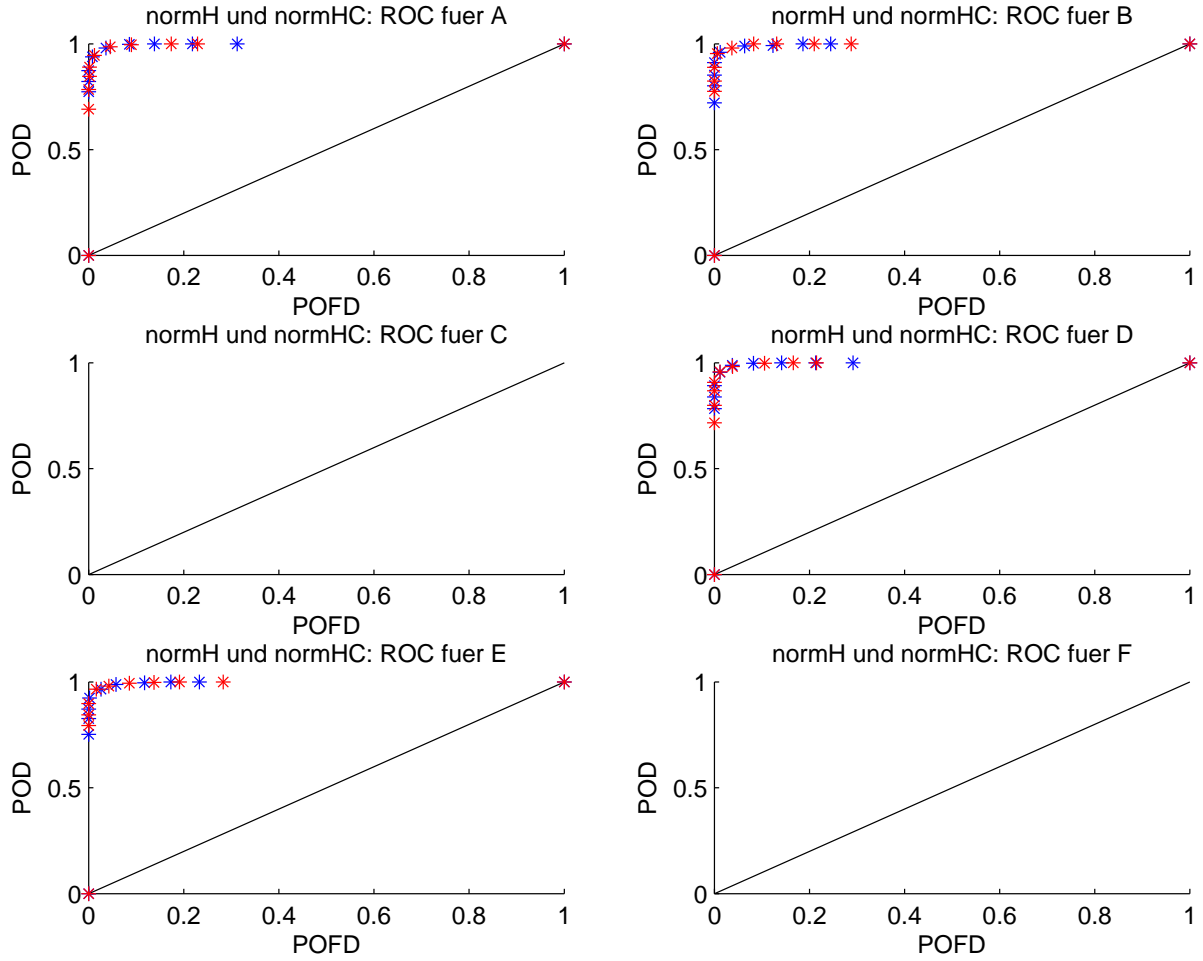


Abbildung 6.34: ROC-Diagramm, H-Norm und HC-Norm

## 6.8 Verwendung anderer Normkonstruktionen

### 6.8.1 Energie-Temperatur-Casimir

Die Norm basierend auf den Energie-Temperatur-Casimir hat gegenüber der Verwendung des Energie-Helizitäts-Casimir den Vorteil, dass sie keine negativen Diskriminanten bewirken kann, solange ein positives  $F$  während der zeitlichen Integration garantiert ist.

Zum Vergleich werden nun die gleichen Auswertungen wie schon für die HC-Norm ( $c1$  von  $-5$  bis  $5$  in Intervallen  $1$ ) gezeigt, verglichen werden nun alle drei Normen (H-Norm, HC-Norm und HS-Norm) miteinander.

## 6 Ergebnisse

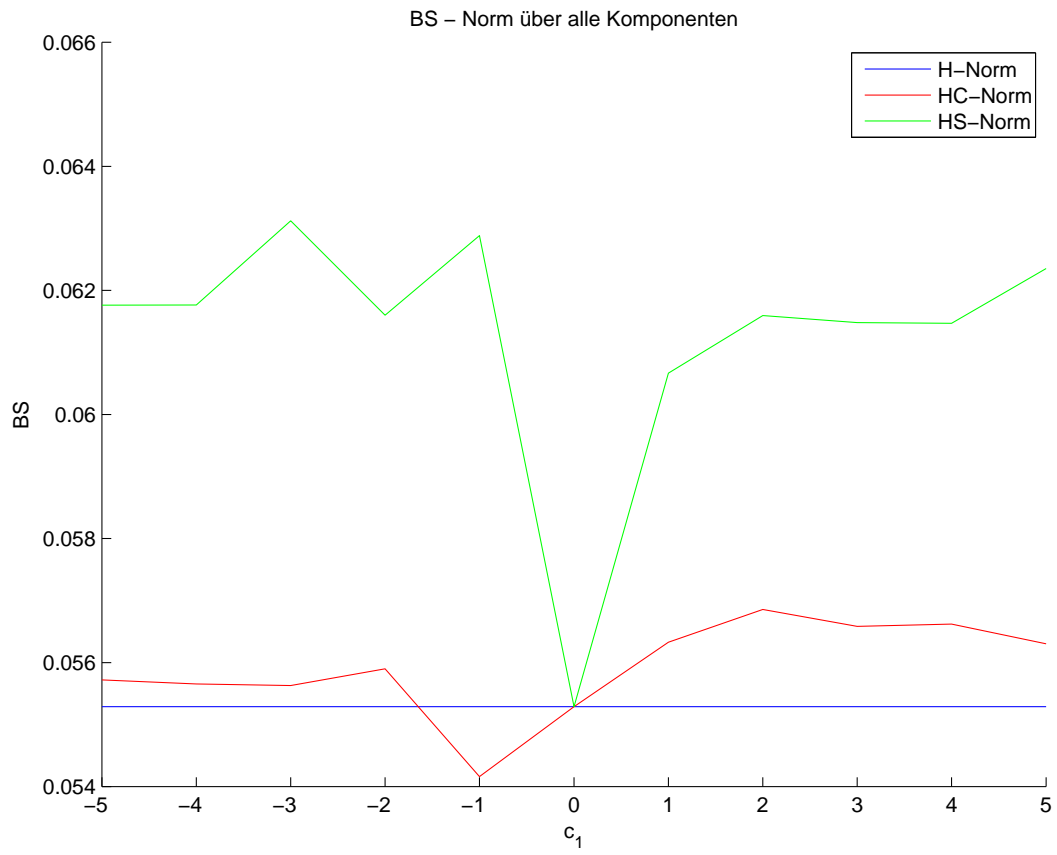


Abbildung 6.35: BS der gesamten Prognose,  $c_1 = [-5, 5]$

## 6 Ergebnisse

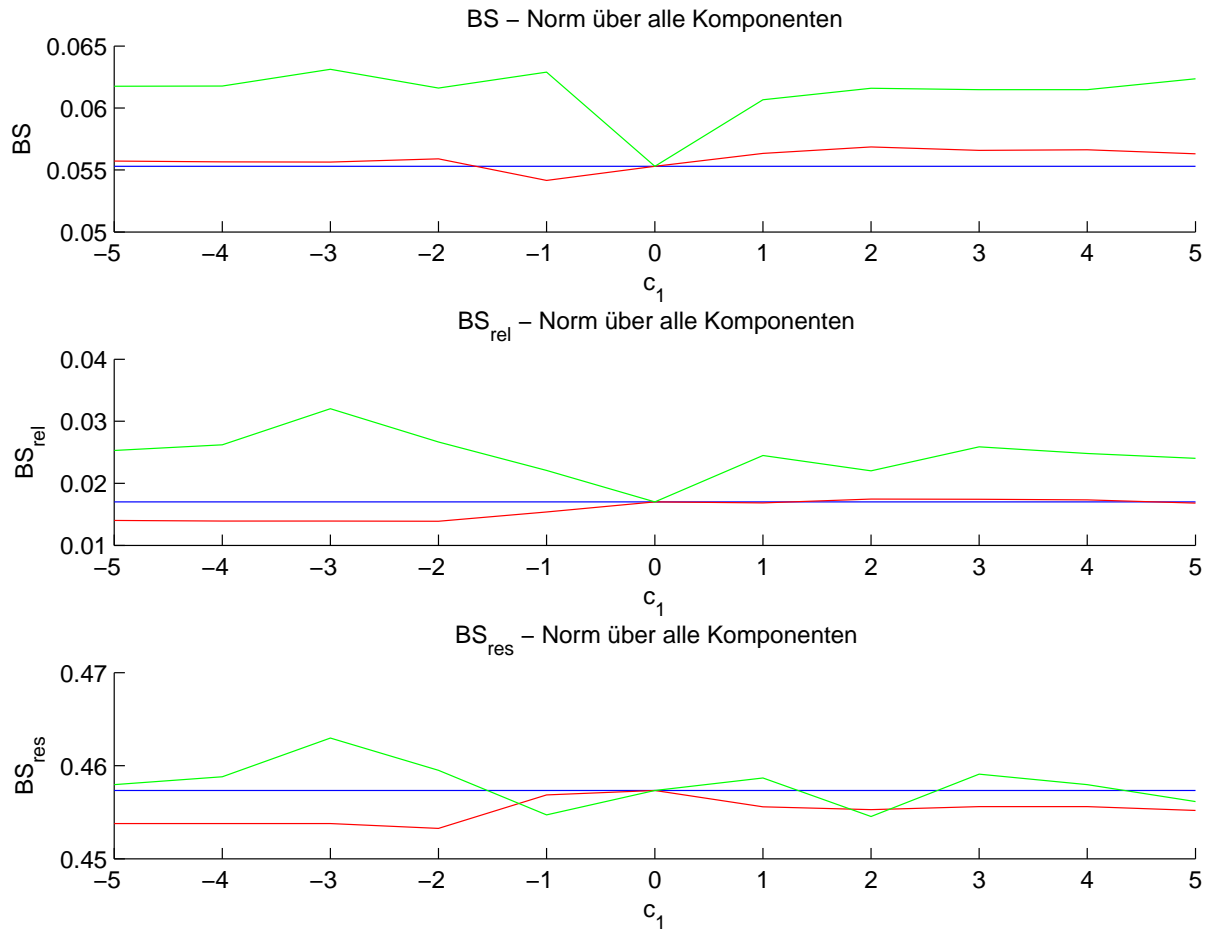


Abbildung 6.36: BS der gesamten Prognose, Aufspaltung in Zuverlässigkeit und Auflösung,  $c_1 = [-5, 5]$

## 6 Ergebnisse

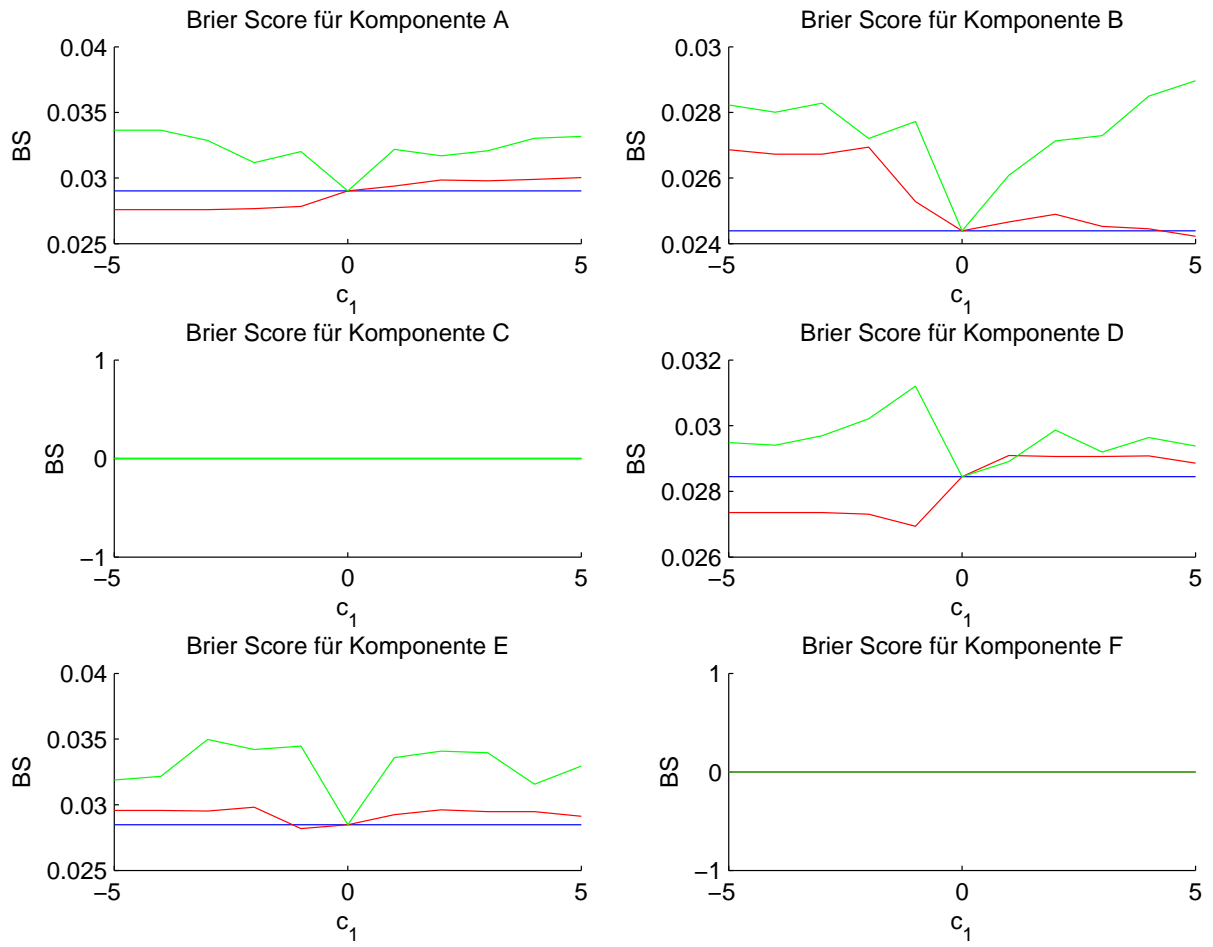


Abbildung 6.37: BS komponentenweise,  $c_1 = [-5, 5]$



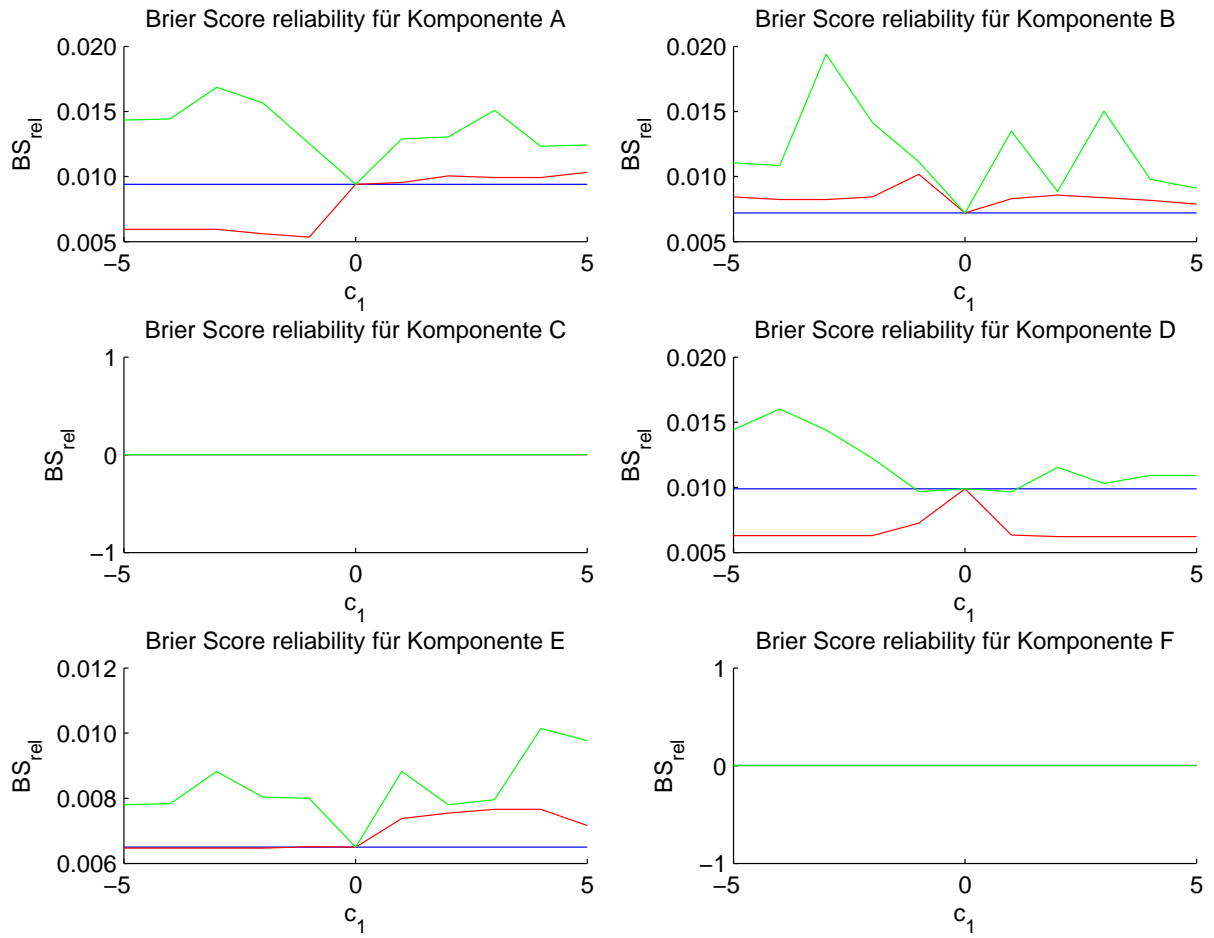
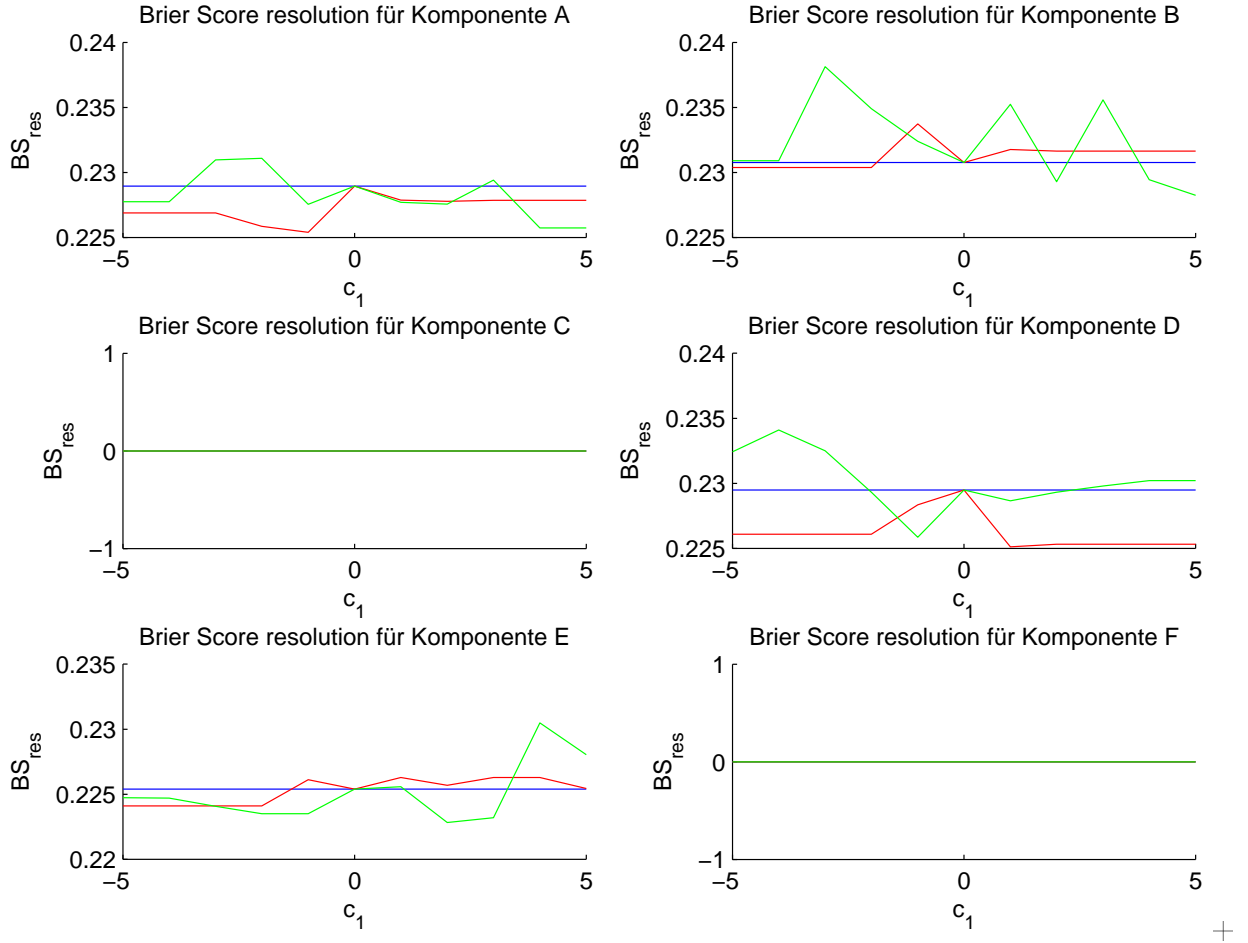


Abbildung 6.38: BS reliability, komponentenweise,  $c_1 = [-5, 5]$

Abbildung 6.39: BS resolution, komponentenweise,  $c_1 = [-5, 5]$ 

Die Verwendung des Energie-Temperatur-Casimirs zeigt in fast allen erdenklichen Darstellungen Verschlechterungen der Brier Scores. Einzig positiv ist zu erwähnen, dass in manchen Wertebereichen die Möglichkeit einer Auflösungsverbesserung besteht, die unter Verwendung der Vorticity als zweiter Casimir ganz ausgeschlossen war.

Unter Miteinbeziehung des Cost-Loss-Verhältnisses des Anwenders könnte versucht werden auch diesen Resultaten etwas Gutes abzugewinnen: Entstehen bei Fehlprognosen keine besonderen Schäden und ist einzig und allein eine schnelle, eindeutige Entscheidung wichtig, so kann auch dieses Konzept zur Auflösungsverbesserung verwendet werden. Ohne der Einstellung „No risk, no fun“ ist es aber weniger empfehlenswert und wird hier nicht mehr näher behandelt.

Allerdings sei schon noch erwähnt, dass die Erhaltung des Casimirs  $\mathcal{S}$  deswegen rein mathematisch nicht weniger wichtig für die dynamische Entwicklung der Systemvariablen ist als die Erhaltung von Gesamtenergie oder Helizitäts-Casimir. Im Anschluss an die auf Seite 115 versuchte Interpretation im Zuge einer unimodalen  $c_1$ -Verteilung, liegt das Problem hier vielleicht genau darin, dass es kein global bevorzugtes  $c_1$  gibt. Somit

scheitert das Konzept einer Linearkombination mit konstantem Koeffizienten.

### 6.8.2 Energie-Helizitäts-Temperatur-Casimir

Auch die Verwendung einer Norm, die sich aus der Linearkombination von Hamilton-Funktion, Helizitäts-Casimir und Temperatur-Casimir zusammensetzt, wurde getestet. Doch was kann nach den Ergebnissen in Kap. 6.8.1 davon noch erwartet werden?

Auch wenn schon erahnt wird, was hierbei herauskommt, wird die folgende Darstellung nicht unterschlagen, da sie einen schön zusammenfassenden Vergleich von

1. H-Norm ( $c_1 = c_2 = 0$ )
2. HC-Norm ( $c_1 = [-5, 5], c_2 = 0$ )
3. HS-Norm ( $c_1 = 0, c_2 = [-5, 5]$ )
4. HCS-Norm ( $c_1, c_2 = [-5, 5]$ )

darstellt.

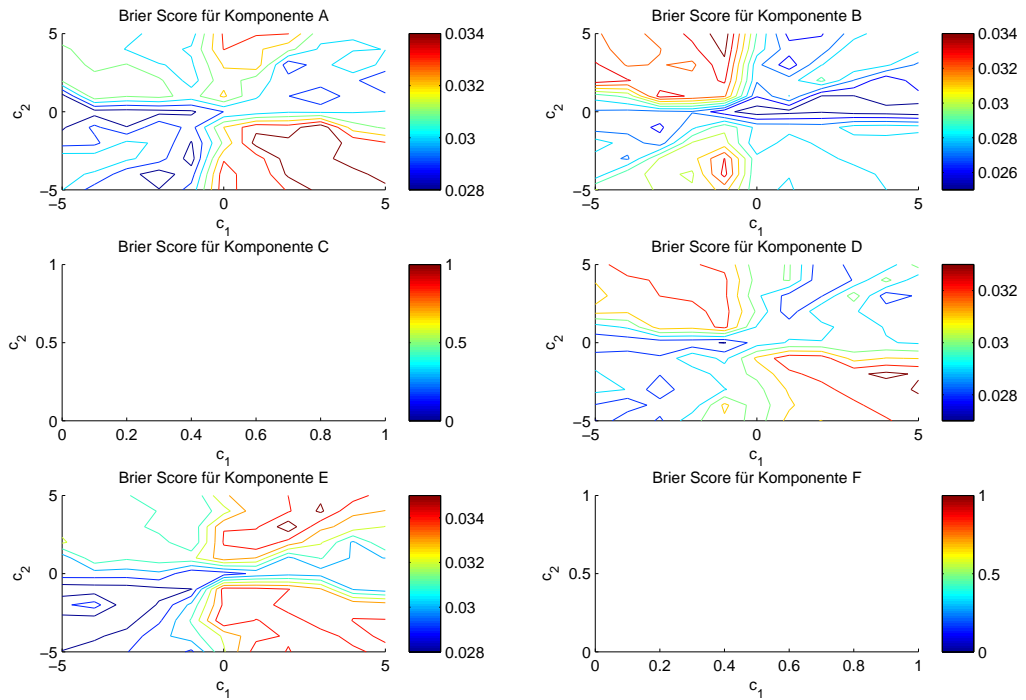


Abbildung 6.40: Contourplot komponentenweise,  $c_1 = [-5, 5]$ ,  $c_2 = [-5, 5]$

Schon die einzelnen Komponenten in Abb. 6.40 zeigen eine Ansammlung der bessern Brier Scores in der Nähe der  $c_2 = 0$ -Werte.

## 6 Ergebnisse

Abb. 6.41 rekonstruiert tatsächlich die Erkenntnis aus Kap. 6.7, die HC-Norm liefert für  $c_1=-1$  den besten Score, auch andere lokale Minima können diesen nicht unterbieten.

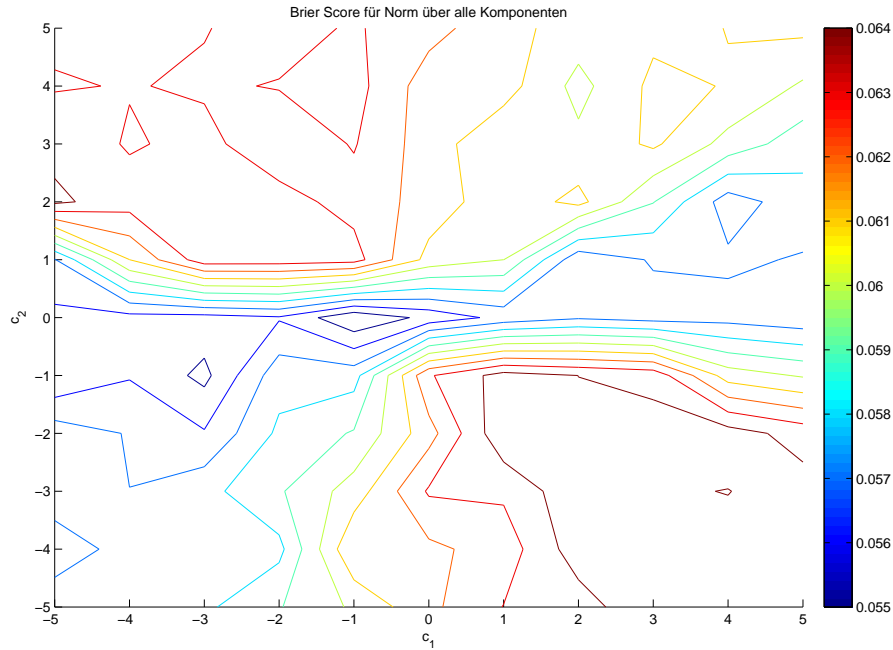


Abbildung 6.41: Contourplot, Norm über alle Komponenten,  $c_1=[-5,5]$ ,  $c_2=[-5,5]$

Das Ergebnis dieses Versuchs noch eine bessere Norm zu generieren führt zur Einsicht, dass bei allen Bemühungen in dem hier festgelegten Rahmen trotzdem die Energie-Vorticity-Norm die besten Resultate erzielt.

Vielleicht wäre es möglich unter feinerer Auflösung von  $c_1$  und  $c_2$  doch noch einen Vorteil für die Verwendung aller drei Erhaltungsgrößen zur Störungsgenerierung zu finden, allerdings steigt der Rechenaufwand für eine Variation von zwei Konstanten quadratisch, außerdem sprechen die Darstellungen in Kap 6.8.1 dagegen.

Der in Kap. 5.2.2.3 auf Seite 76 geäußerte Wunsch, die gesamte Geometrie des Attraktors durch die Gewichtungsmatrizen zu erfassen setzt zwar sehr wohl die Verwendung aller Erhaltungsgrößen voraus, diesem kann jedoch nicht mehr im Zuge dieser Arbeit und nicht unter Verwendung eines linearen Konzepts weiter nachgegangen werden.

Die physikalische Interpretation und Übertragbarkeit eines Energie-Vorticity-Casimirs auf die reale Atmosphäre ist in Anlehnung an das Konzept der Erhaltung von Gesamtenergie und Ertelscher potentieller Vorticity, sehr naheliegend und lässt die Frage nach der Bedeutung des Temperatur-Casimirs unter „realen“ Verhältnissen für weitere Betrachtungen offen.

## 7 Conclusio und Ausblick

Das reduzierte Gleichungssystem der Rayleigh-Bénard-Konvektion von Saltzman (1957) wurde zu Test- und Lehrzwecken benutzt um theoretische Grundlagen für die Optimierung höher-dimensionaler Modelle zu schaffen.

Der Erfolg des untersuchten Konzepts kann mit folgenden Grafiken aus Kap. 6 dargestellt werden:

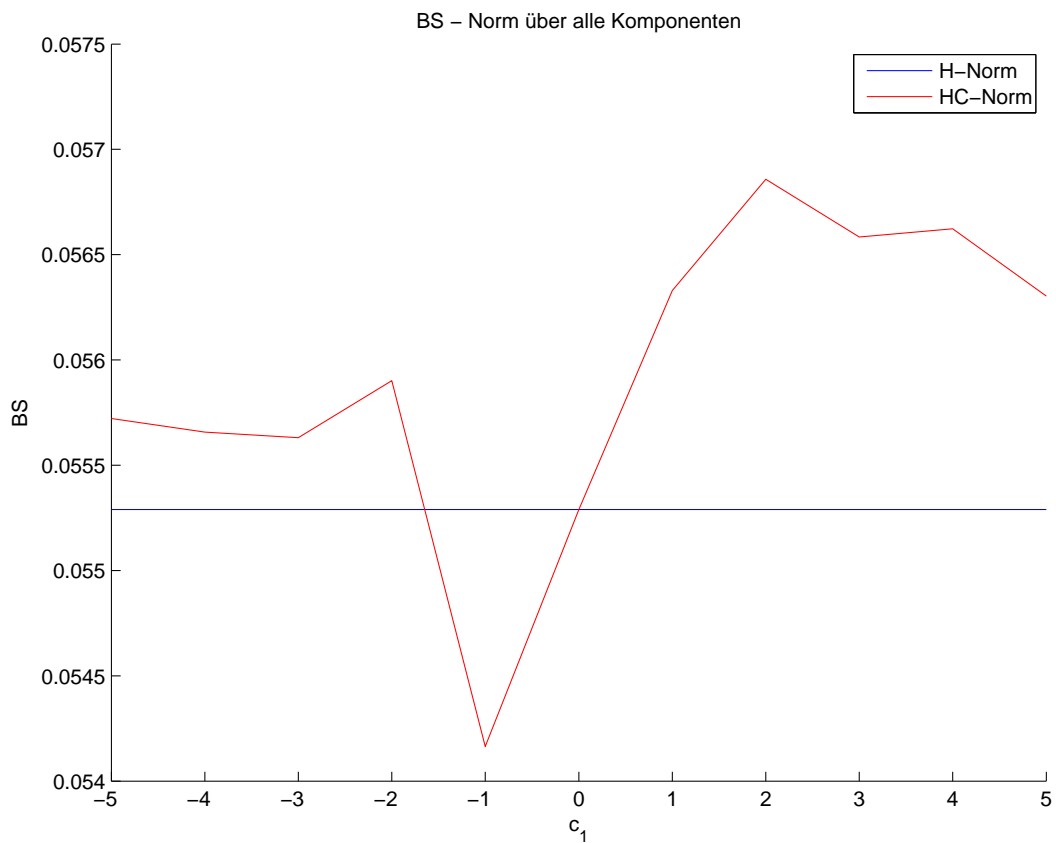


Abbildung 7.1: BS der gesamten Prognose

## 7 Conclusio und Ausblick

**c1=-1.1**

	BS	BS_rel	BS_res
Norm über alle Komp.	-	-	-

Abbildung 7.2: Verbesserung der Brier Scores in Zuverlässigkeit und Auflösung gegenüber der Energienorm für  $c1=-1.1$

**c1=-1.1**

	BS	BS_rel	BS_res
Strömungs- Feld			
Temperatur- Feld			

Abbildung 7.3: Verbesserungen der physikalischen Feldprognosen gegenüber der Energienorm für  $c1=-1.1$

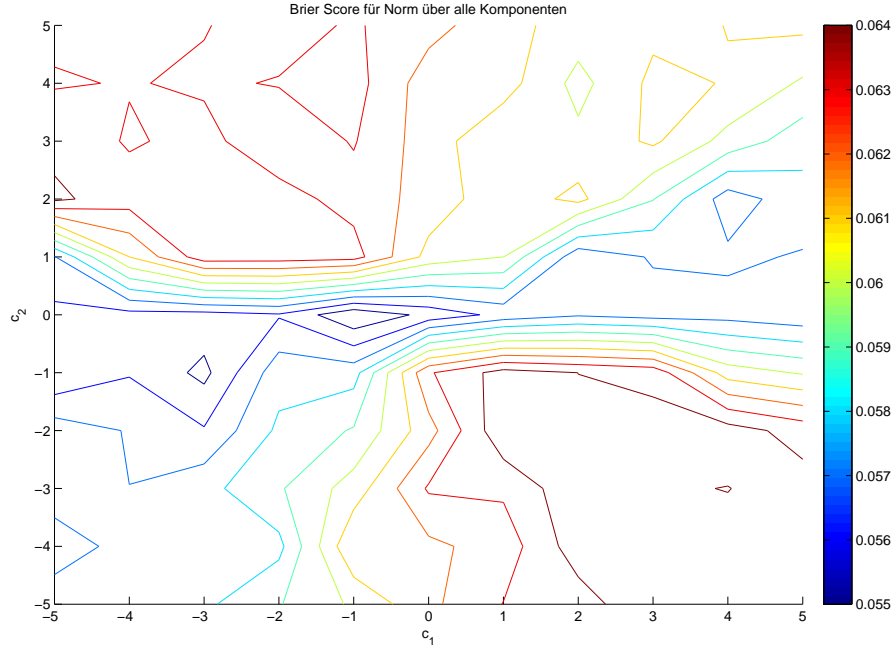


Abbildung 7.4: Contourplot, Norm über alle Komponenten,  $c_1=[-5,5]$ ,  $c_2=[-5,5]$

Da nun gezeigt wurde, dass es prinzipiell möglich ist, ein EPS durch Einführung von Energie-Vorticity-Casimiren in die Störungsrechnung zu optimieren, kann dieses Konzept auf viele weitere Modelle bis hin zur tatsächlichen Anwendung ausgebaut werden.

Nachdem sogar im hier implementierten, einfachen System noch zahlreiche Parameter existieren, unter deren Variation noch weitere Versuche durchgeführt werden könnten, ist gar nicht auszudenken, wie viele Möglichkeiten komplexere Prognosemodelle bieten.

Es könnte unter anderem noch ausgetestet werden, wie sich die erzielte Verbesserung unter Veränderung der Ensemblememberanzahl verhält. Bringen neue Normen schon Vorteile bei weniger Ensembleläufen, so könnte die ersparte Rechenleistung in andere Parameter, wie zeitlich oder räumlich höhere Auflösung investiert werden.

Auch eine theoretischere Herangehensweise an die Konstruktion neuer Normen wäre denkbar. Durch genauere Analysen der Erhaltungsgrößen bezüglich deren geometrischen Verhaltens zueinander brächten vielleicht neue Erkenntnisse. Möglicherweise ließen sich anhand näherer Betrachtung der Schnittwinkeln der Casimirflächen im Phasenraum andere Gewichtungsfaktoren, unter Umständen auch nichtlineare Kombinationen der Normen finden, die die Resolvente  $A$  noch besser modifizieren könnten.

Ist der Systemattraktor bekannt, so könnten Funktionen  $c_i = c_i(A, B, C, D, E, F)$  ermittelt werden, die alle Erhaltungsgrößen zum Analysezeitpunkt in Abhängigkeit des jeweils aktuellen Zustands speziell gewichten. Diese Funktionen könnten entweder ana-

lytisch durch Minimierung einer Kostenfunktion oder explizit durch Unterteilung des Attraktors in genügend viele Regionen mit Angabe bestimmter Werte konstruiert werden.

Zwar wäre die Berechnung dieser Funktionen wahrscheinlich sehr aufwändig, sie könnten in der operationellen Vorhersage dann jedoch uneingeschränkt eingesetzt werden und würden die optimale Nutzung der Kenntnis diverser konservativen Größen für alle Prognosen garantieren.

Die Auswirkungen der Normen könnten aber auch noch etwas früher abgegriffen werden, wenn die generierten Störungen testweise nichtlinear integriert und statistische Untersuchungen bezüglich ihrer Entfernung vom Attraktor angestellt würden. Somit könnte herausgefunden werden, bei Verwendung welcher Normen am ehesten Störungen für tatsächlich mögliche Endzustände generiert werden.

Hier wurde zur Verifikation erst die gesamte Vorhersagewahrscheinlichkeit, prozentuell gegeben durch alle Ensemblemembers, verwendet.

Die Grundlage zur Entscheidung, ob die Vorhersage nun gut oder weniger günstig war besteht nur in der Separation der Zustände in positiv und negativ. Diese Einteilung ist zwar sinnvoll für aperiodisch schwankende Entwicklungskoeffizienten, allerdings trotzdem relativ ungenau. Liefert die Beobachtung einen Wert von beispielsweise  $+0.1$ , so wird diese bei einer Prognose von  $+5$  besser bewertet, als eine Vorhersage von  $-0.1$ . Eine Möglichkeit dieser Ungenauigkeit entgegenzuwirken wären feinskaligere Verifikationsintervalle.

Es kann zusammenfassend gesagt werden, dass dieses Konzept der kombinierten Normen entweder

- direkt die Prognosequalität steigern kann, oder
- Rechenleistung, die anderweitig eingesetzt zu noch größeren Verbesserungen führen könnte, einsparen kann.

Diese Aussichten gewinnen umso mehr an Bedeutung, da derzeit unter anderem auch am ECMWF standardmäßig reine Energienormen (unter Einbeziehung der Feuchte in speziellen Gebieten) verwendet werden. Die Möglichkeit der Realisierung des vorgestellten Konzepts in näherer Zukunft besteht tatsächlich, weil andere Erhaltungsgrößen, wie die potentielle Enstrophie berechnet werden und als Norm dienen können (siehe z.B. [14])

Der Traum eines jeden Theoretikers, der sich mit atmosphärischer Physik beschäftigt, ist wahrscheinlich eine 100% deterministische und korrekte Zustandsanalyse und -prognose. Aufgrund der terrestrischen Komplexität und Einflüssen mit Ursprung auch außerhalb der Atmosphäre, aufgrund zahlreicher physikalischer Interaktionen und nicht exakt lösba-



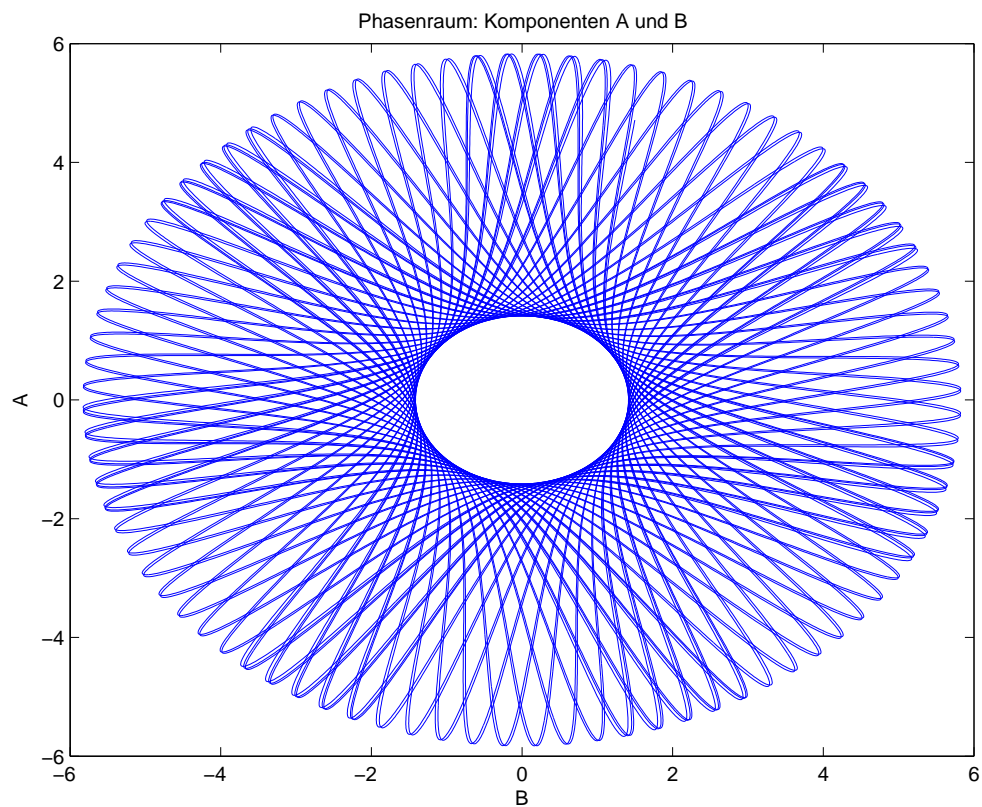
rer Vorhersagegleichungen und schließlich auch aufgrund des jederzeit möglichen Flügel-schlags eines Schmetterlings, der alles verändern könnte, ist die Erfüllung dieses Traums in nächster Zeit eher unwahrscheinlich.

Doch genau aus diesem Grund sollte der Ensemblevorhersage, die eine gewisse Fehlbarkeit der Modelle zugibt, gebührende Aufmerksamkeit geschenkt werden.

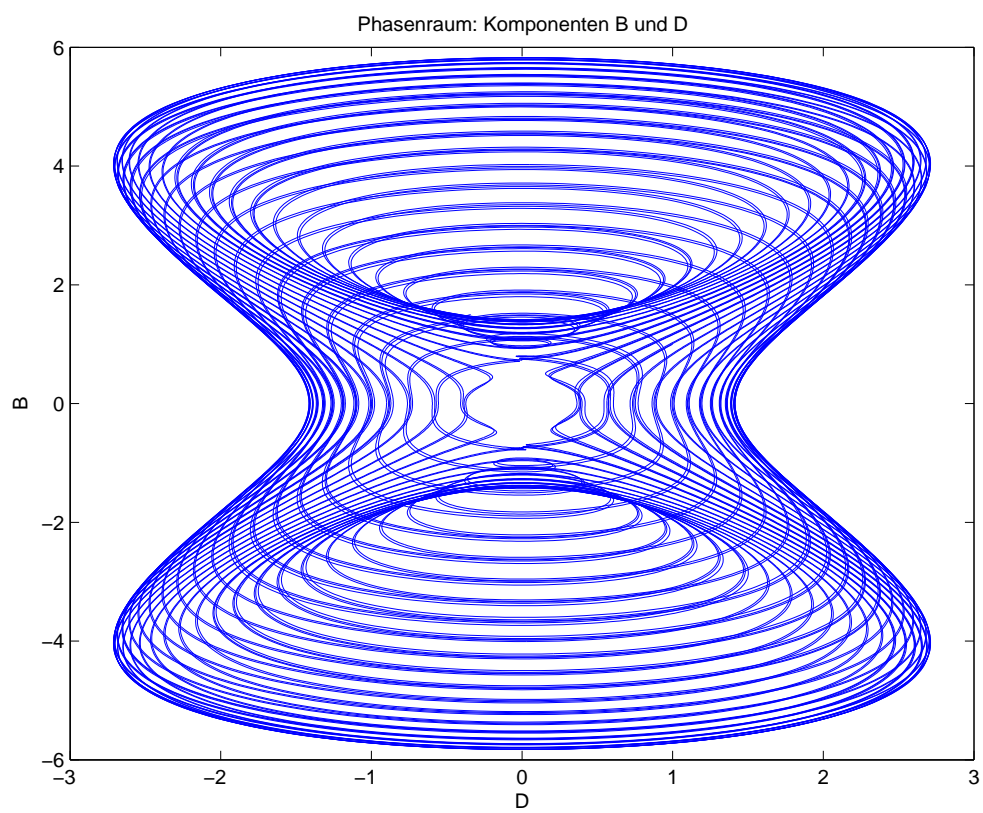
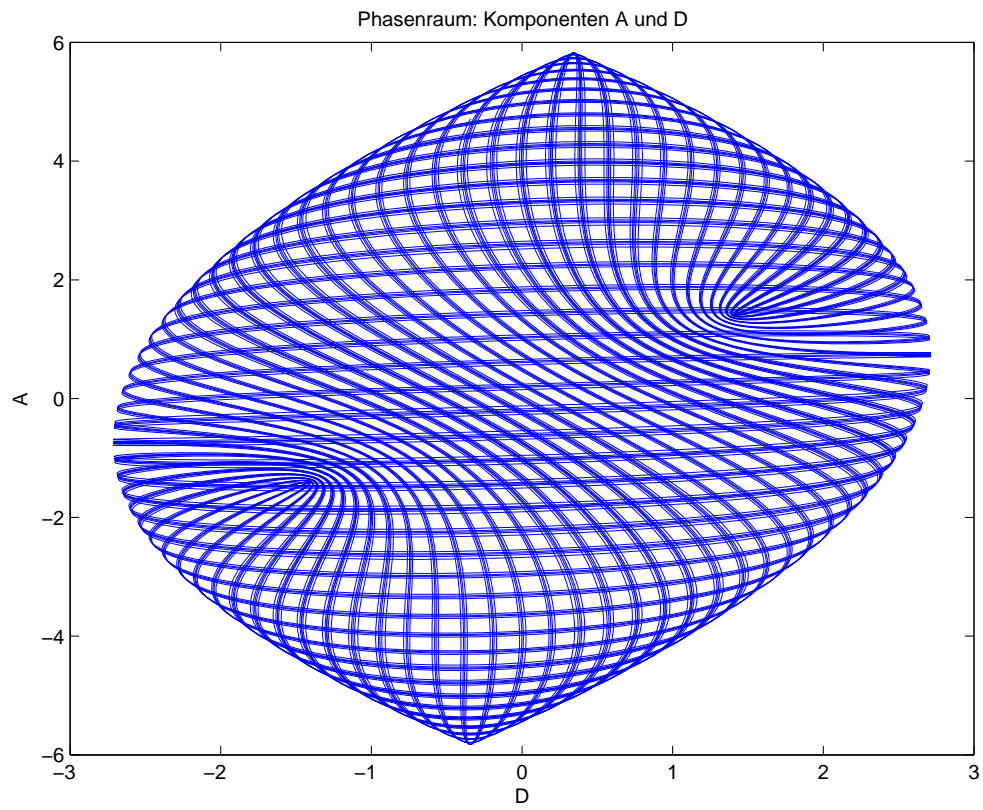
## **Teil II: Anhang**

# Abbildungen

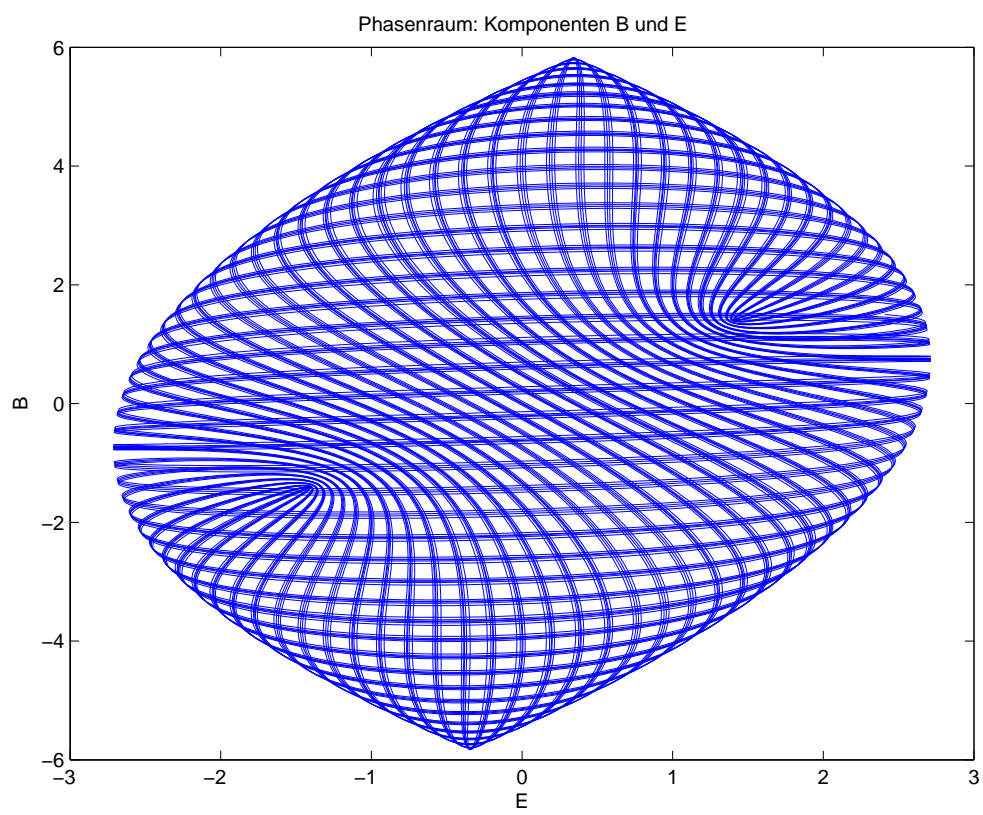
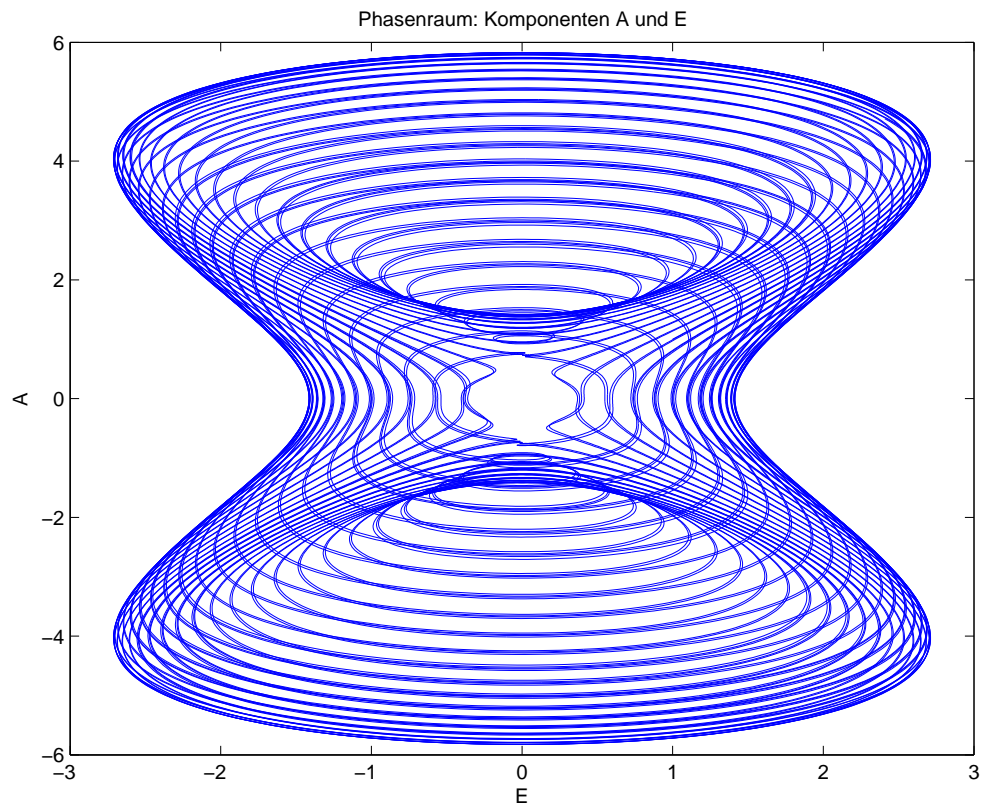
## Phasenraumplots 2-dimensional



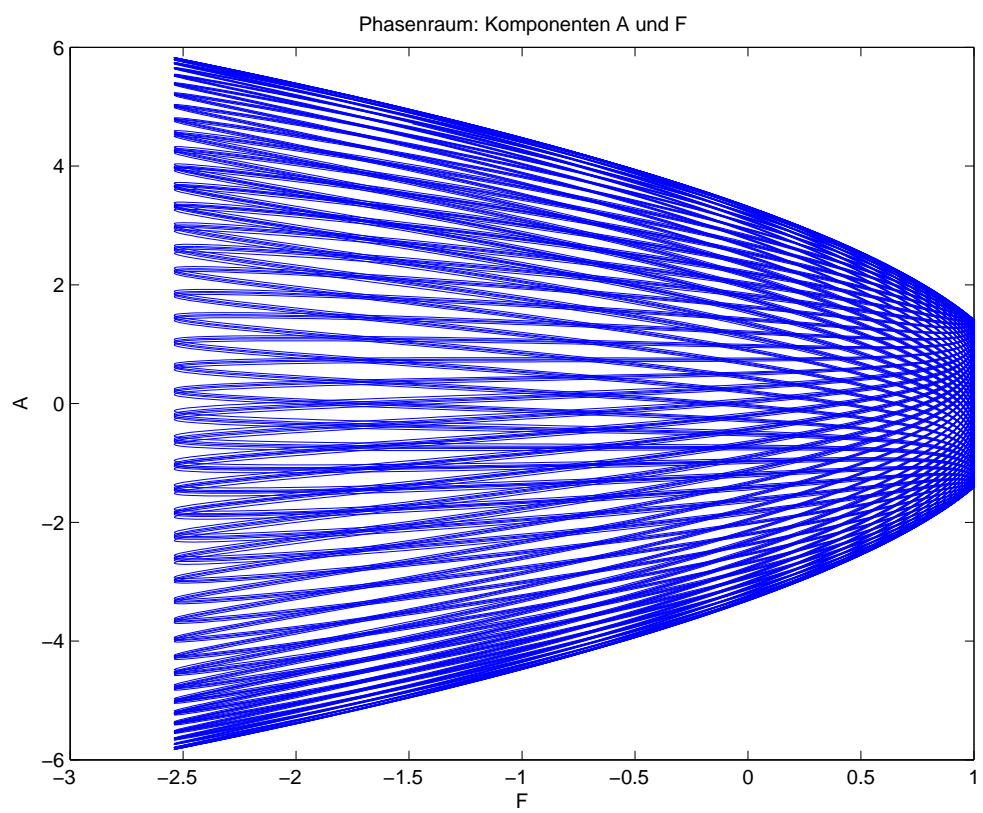
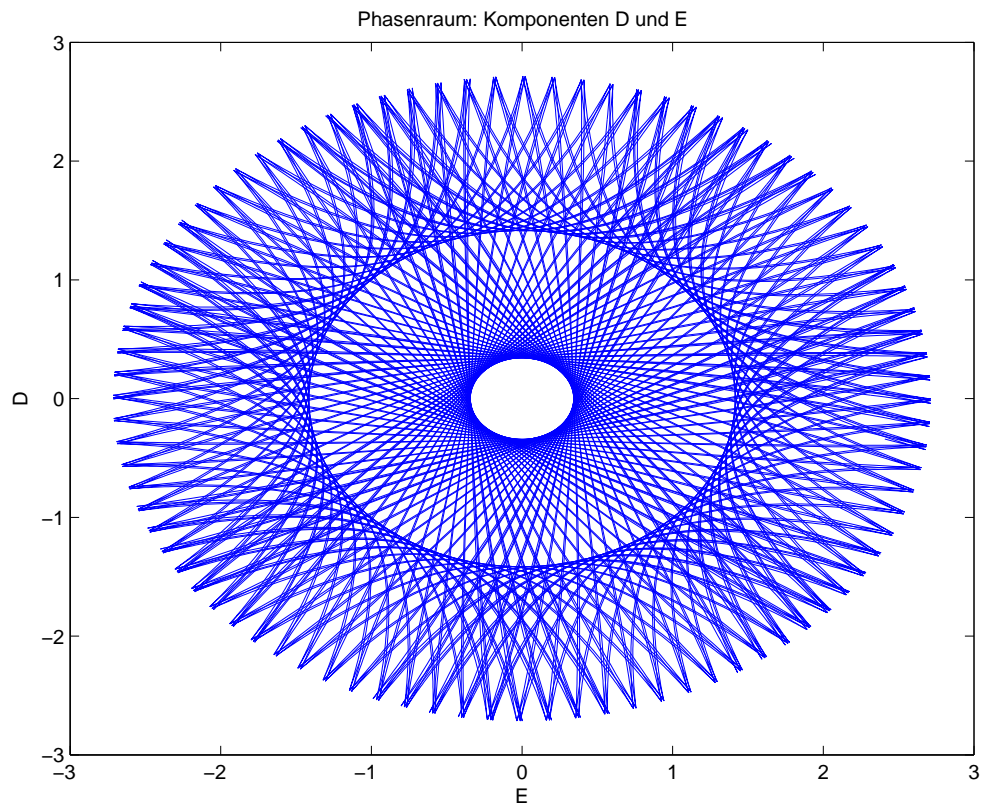
## Abbildungen



## Abbildungen

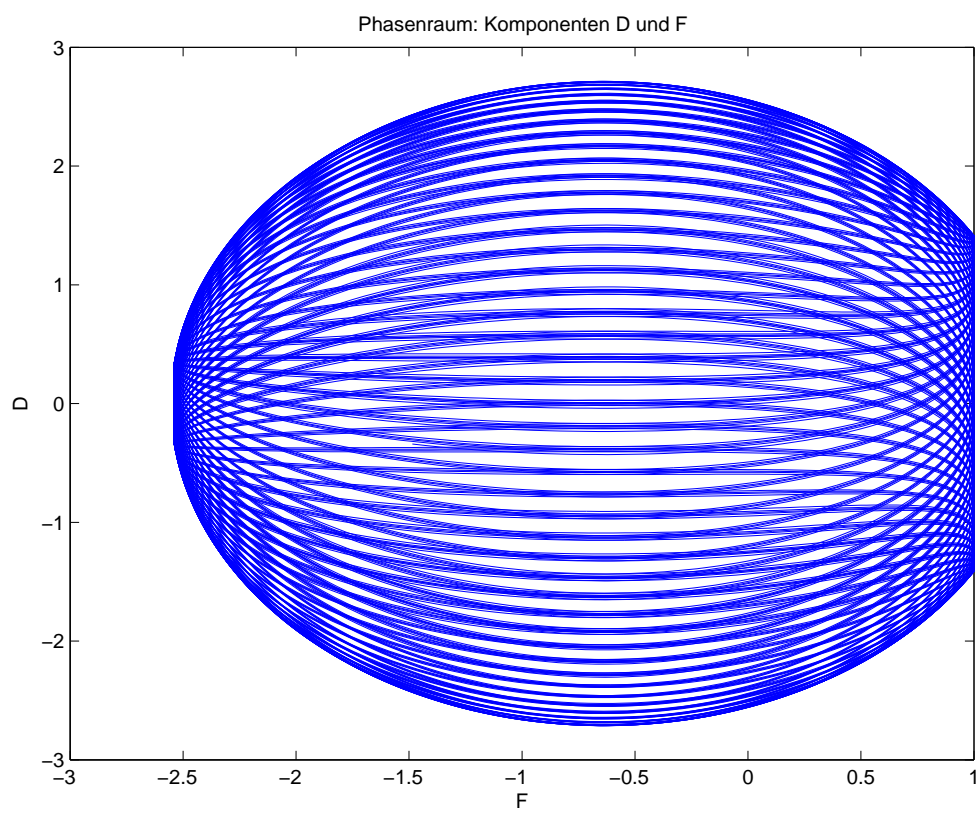
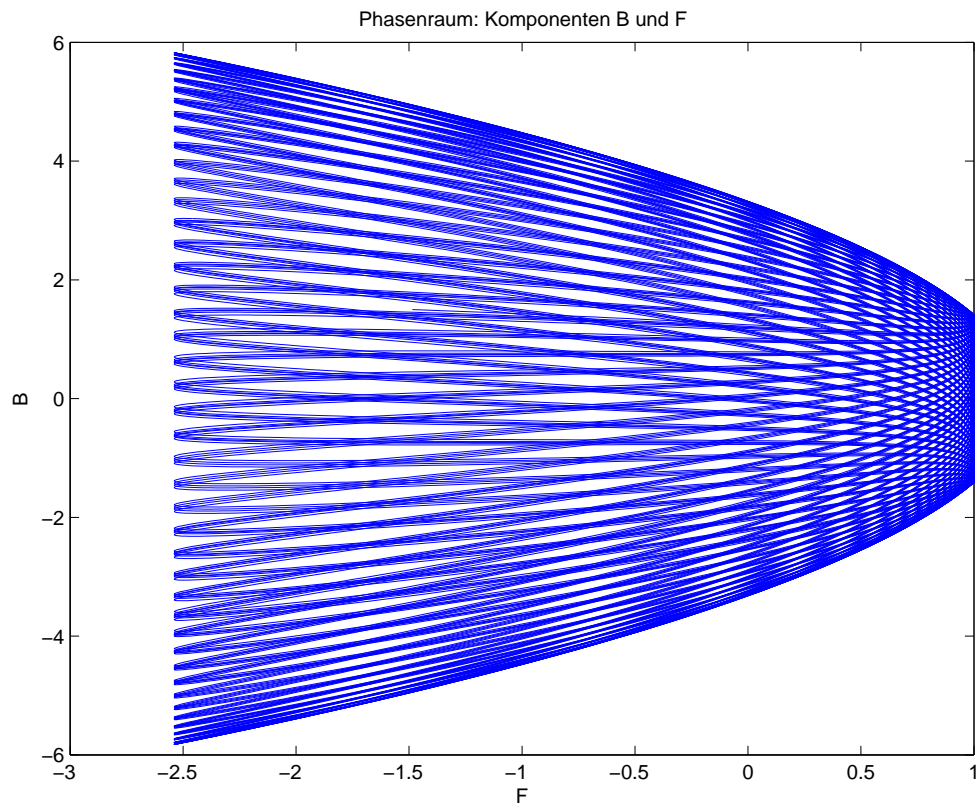


## Abbildungen

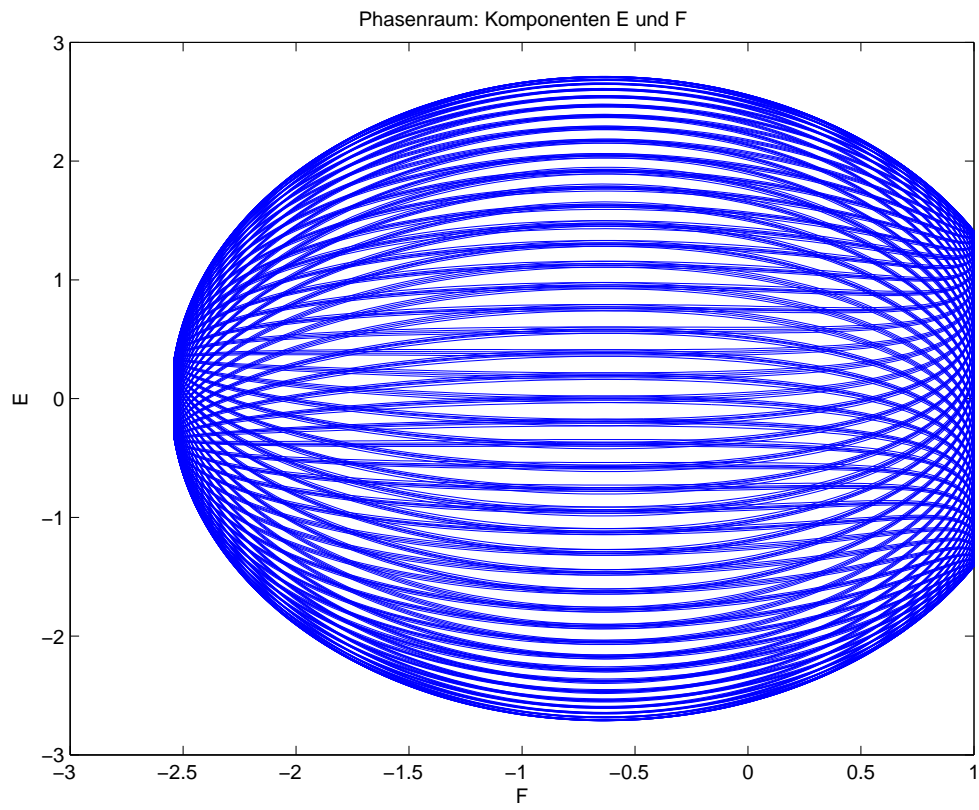




## Abbildungen

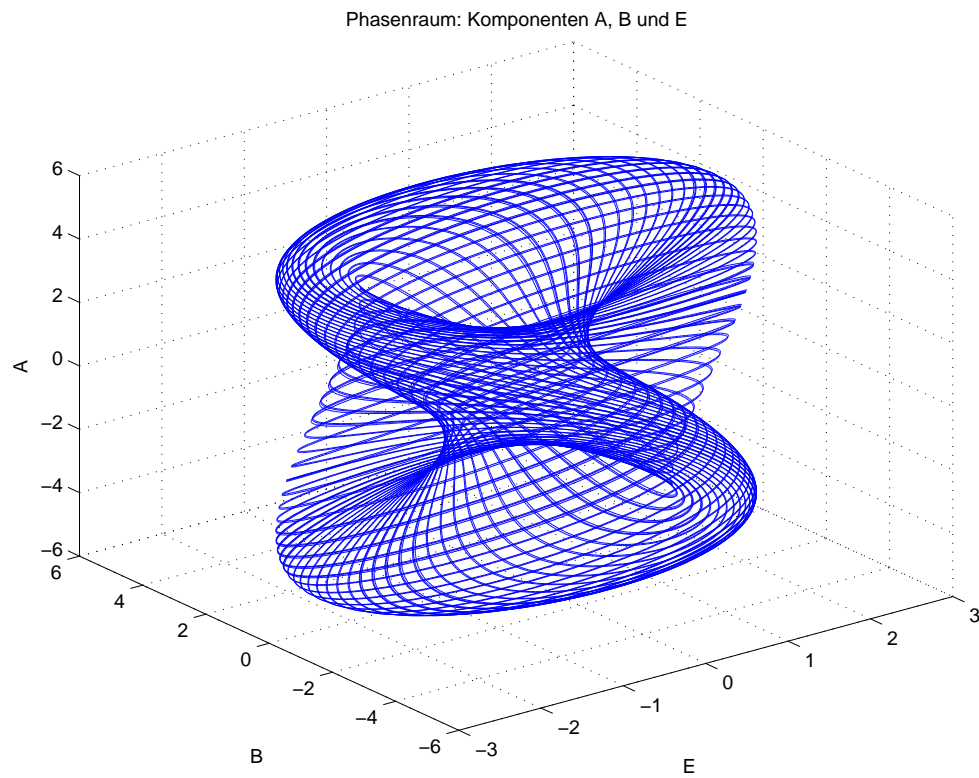
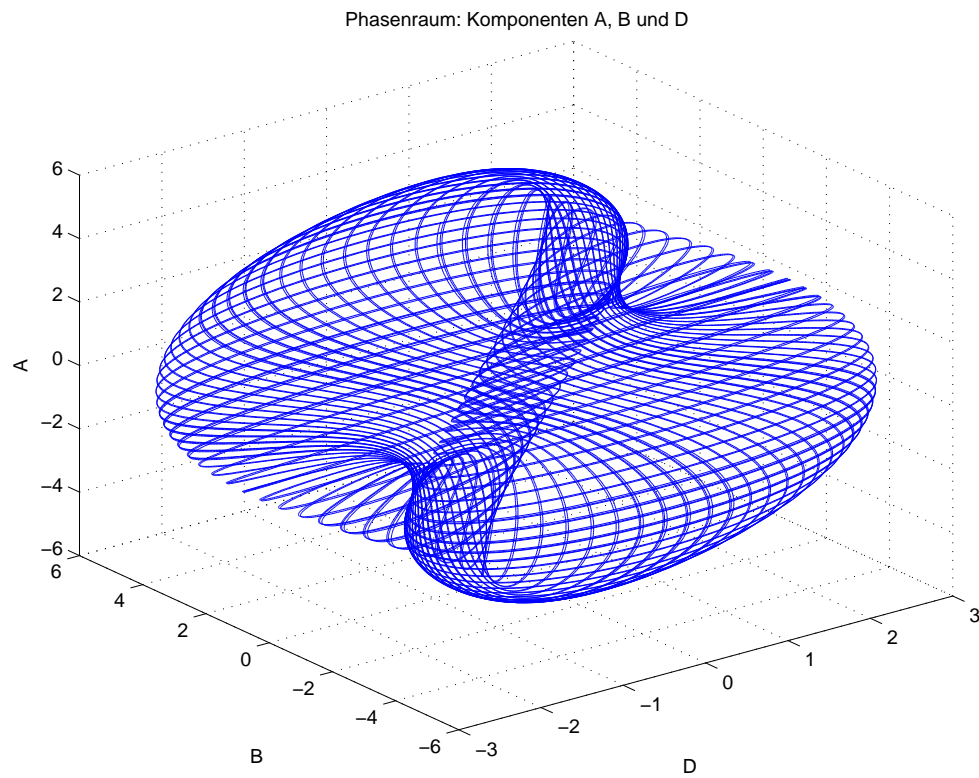


## Abbildungen



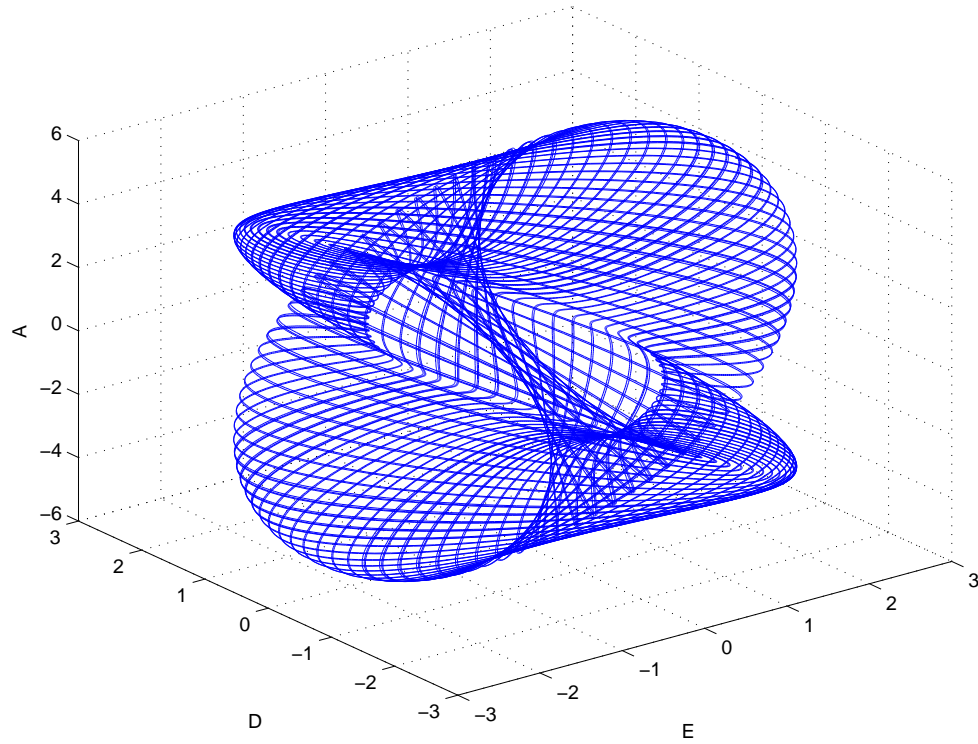


## Phasenraumplots 3-dimensional

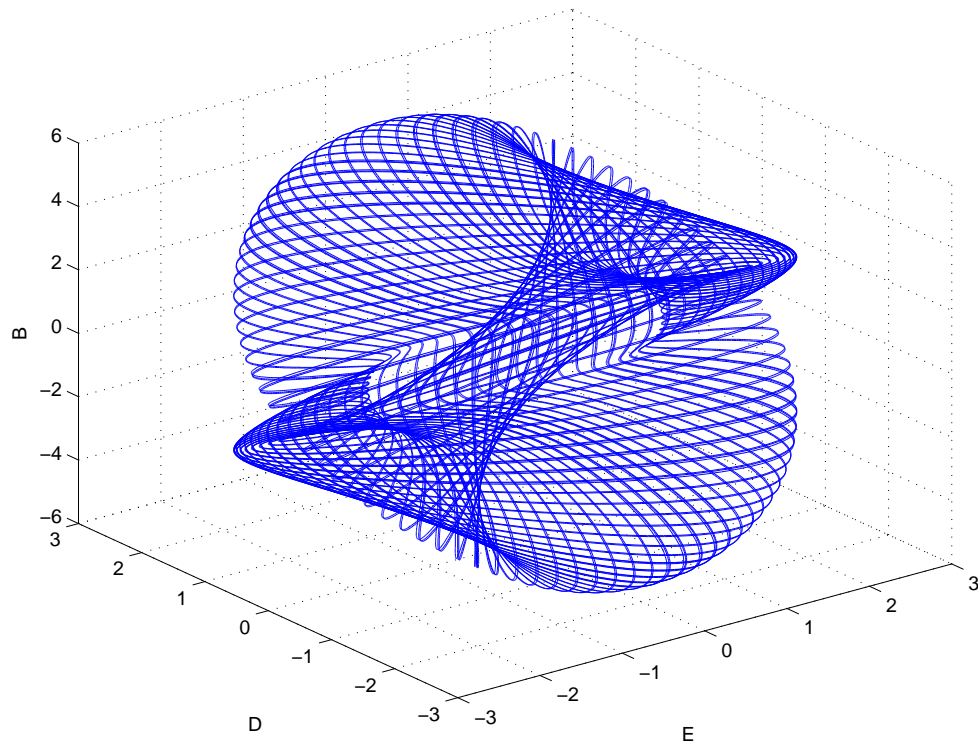


## Abbildungen

Phasenraum: Komponenten A, D und E

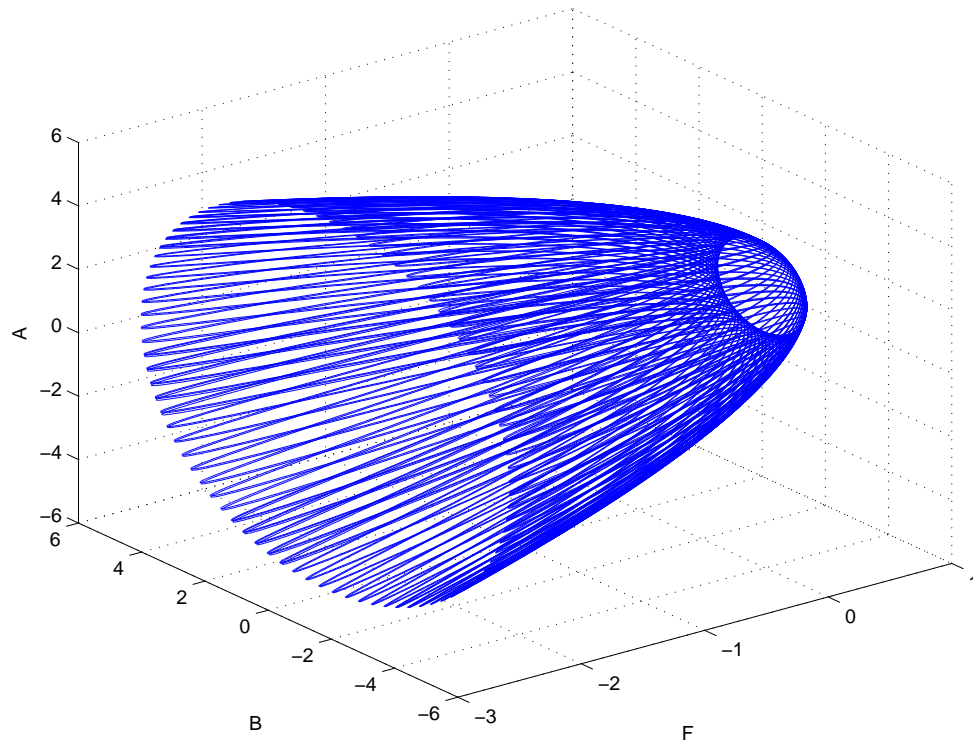


Phasenraum: Komponenten B, D und E

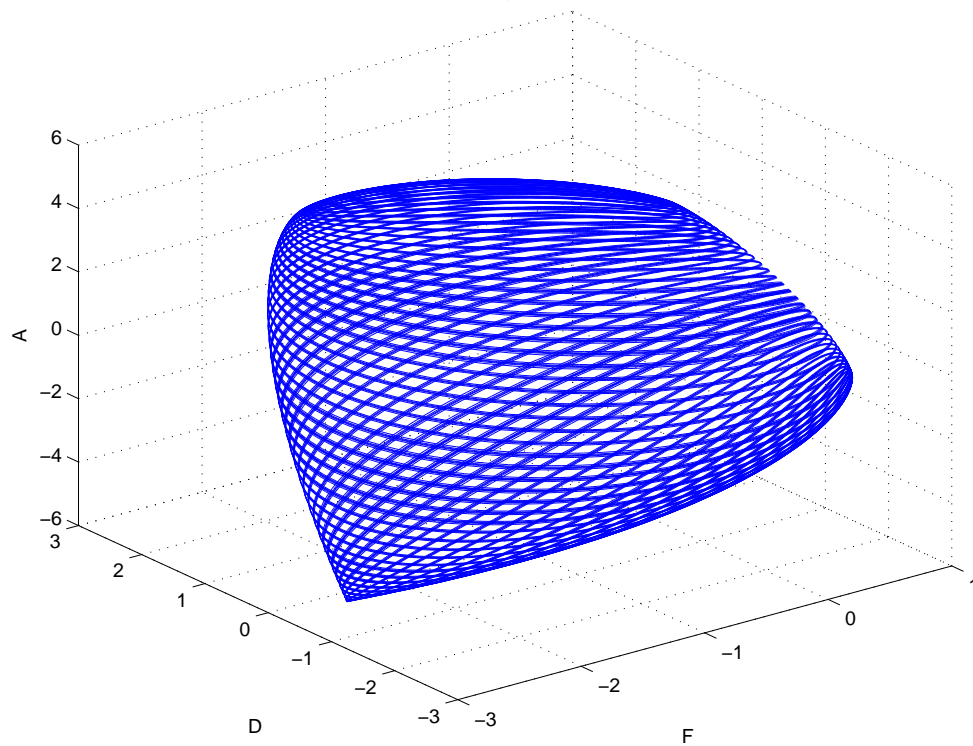


## Abbildungen

Phasenraum: Komponenten A, B und F

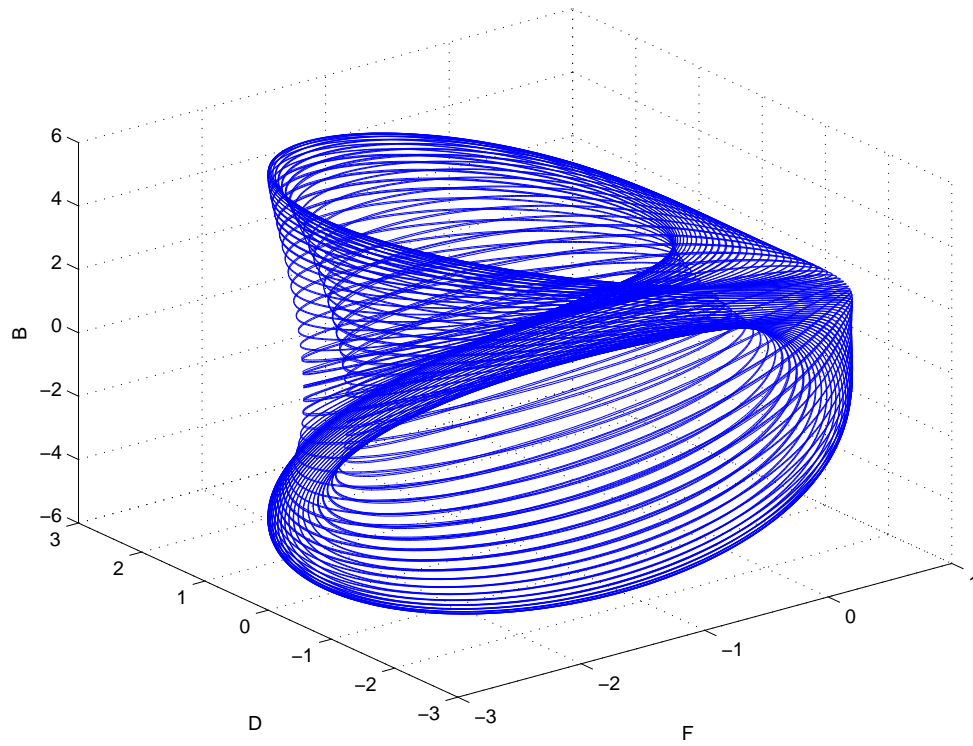


Phasenraum: Komponenten A, D und F

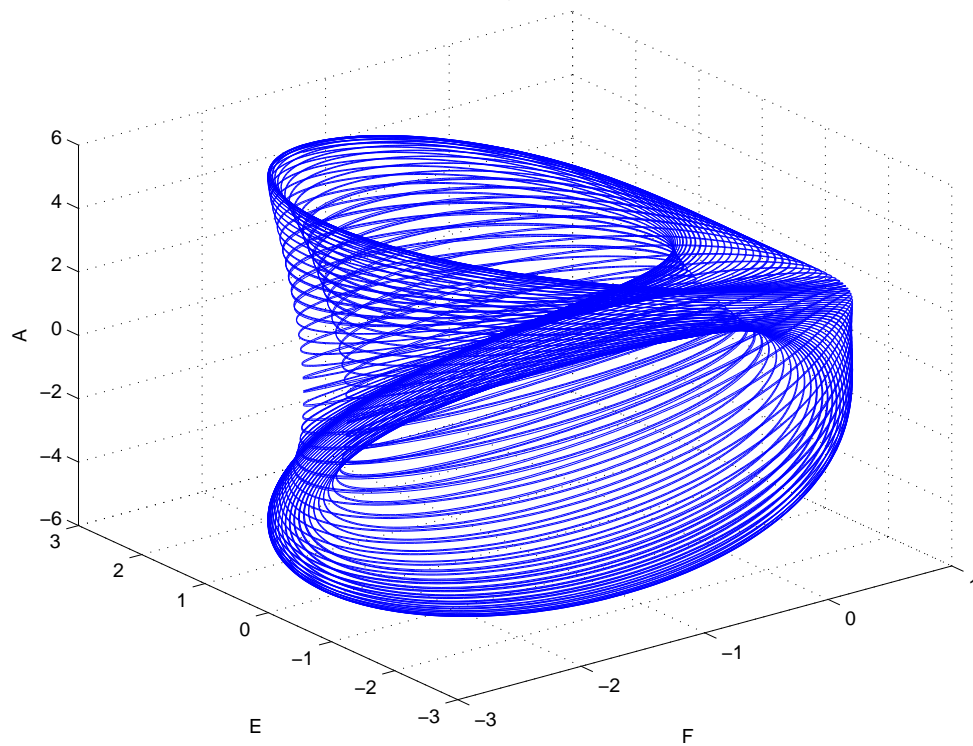


## Abbildungen

Phasenraum: Komponenten B, D und F

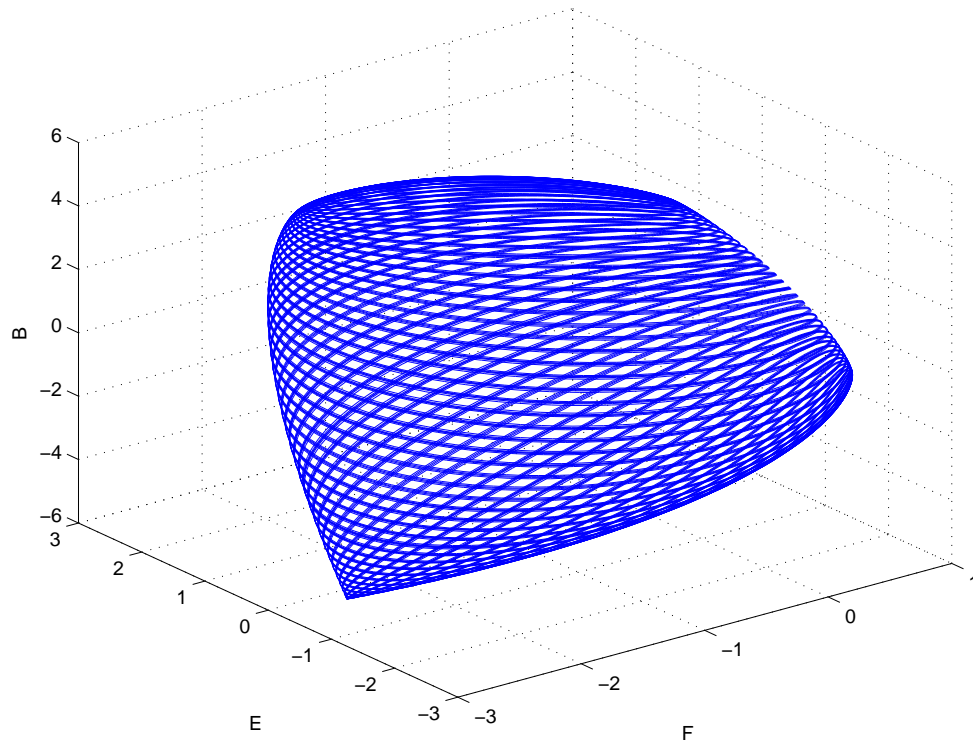


Phasenraum: Komponenten A, E und F

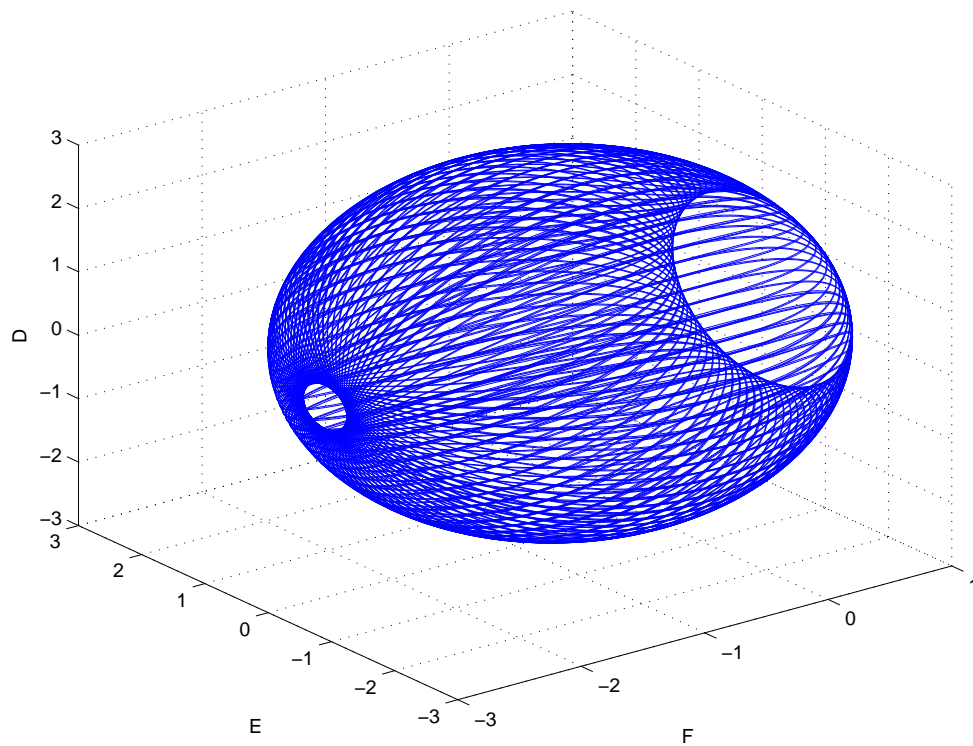


## Abbildungen

Phasenraum: Komponenten B, E und F



Phasenraum: Komponenten D, E und F



# Txt-Outputs

**c1=-1**

```
1  -----Parameterwahl-----
2  dt= 0.001
3  t_end= 20
4  dt_ana= 0.100
5  n=30
6  c1= -1.000
7  c2= 0.000
8  emp_fac= 1.000
9
10 -----Verifikation-----
11
12 Norm 1 = H-Norm
13 Norm 2 = H-Norm + c1 * C-Norm
14 Norm 3 = H-Norm + c1 * C-Norm + c2 * S-Norm (kann für c2=0 ignoriert werden.
15
16
17      Norm 1      Norm 2      Norm 3
18
19 Diff zu norm1: 1 bedeutet besser,0 gleich und -1 schlechter als norm1
20
21 fc_step= 0.100
22
23      0.00      1.00      1.00
24      0.00     -1.00     -1.00
25      0.00      0.00      0.00
26      0.00      1.00      1.00
27      0.00      1.00      1.00
28      0.00      0.00      0.00
29
30 BS
```

```

31
32 fc_step= 0.100
33
34     0.029011725    0.027833613    0.027833613
35     0.024388610    0.025281965    0.025281965
36     0.000000000    0.000000000    0.000000000
37     0.028447795    0.026934673    0.026934673
38     0.028481295    0.028185371    0.028185371
39     0.000000000    0.000000000    0.000000000
40
41
42 BS_rel
43
44 fc_step= 0.100
45
46     0.009409548    0.005347571    0.005347571
47     0.007198492    0.010171692    0.010171692
48     0.002500000    0.002500000    0.002500000
49     0.009886935    0.007254726    0.007254726
50     0.006252094    0.006274428    0.006274428
51     0.002500000    0.002500000    0.002500000
52
53
54 BS_res
55
56 fc_step= 0.100
57
58     0.228962021    0.225402557    0.225402557
59     0.230772582    0.233745781    0.233745781
60     0.000000000    0.000000000    0.000000000
61     0.229490417    0.228365746    0.228365746
62     0.225402557    0.226128409    0.226128409
63     0.000000000    0.000000000    0.000000000
64
65
66 BS_unc
67
68 fc_step= 0.100
69
70     0.24969        0.24969        0.24969

```

*Txt-Outputs*

71	0.24999	0.24999	0.24999
72	0.00000	0.00000	0.00000
73	0.24984	0.24984	0.24984
74	0.24969	0.24969	0.24969
75	0.00000	0.00000	0.00000



## c1=-1.1

```

1  -----Parameterwahl-----
2  dt= 0.001
3  t_end= 20
4  dt_ana= 0.100
5  n=30
6  c1= -1.100
7  c2= 0.000
8  emp_fac= 1.000
9
10 -----Verifikation-----
11
12 Norm 1 = H-Norm
13 Norm 2 = H-Norm + c1 * C-Norm
14 Norm 2 = H-Norm + c1 * C-Norm + c2 * S-Norm (kann für c2=0 ignoriert werden.
15
16
17      Norm 1      Norm 2      Norm 3
18
19 Diff zu norm1: 1 bedeutet besser,0 gleich und -1 schlechter als norm1
20
21 fc_step= 0.100
22
23      0.00      1.00      1.00
24      0.00     -1.00     -1.00
25      0.00      0.00      0.00
26      0.00      1.00      1.00
27      0.00      1.00      1.00
28      0.00      0.00      0.00
29
30 BS
31
32 fc_step= 0.100
33
34      0.029011725  0.027666108  0.027666108
35      0.024388610  0.024969291  0.024969291
36      0.000000000  0.000000000  0.000000000
37      0.028447795  0.026934673  0.026934673
38      0.028481295  0.028112786  0.028112786
39      0.000000000  0.000000000  0.000000000

```

```

40
41
42 BS_rel
43
44 fc_step= 0.100
45
46     0.009409548    0.005431323    0.005431323
47     0.007198492    0.011911342    0.011911342
48     0.002500000    0.002500000    0.002500000
49     0.009886935    0.007254726    0.007254726
50     0.006252094    0.006475433    0.006475433
51     0.002500000    0.002500000    0.002500000
52
53
54 BS_res
55
56 fc_step= 0.100
57
58     0.228962021    0.225988822    0.225988822
59     0.230772582    0.235887442    0.235887442
60     0.000000000    0.000000000    0.000000000
61     0.229490417    0.228365746    0.228365746
62     0.225402557    0.226128409    0.226128409
63     0.000000000    0.000000000    0.000000000
64
65
66 BS_unc
67
68 fc_step= 0.100
69
70     0.24969        0.24969        0.24969
71     0.24999        0.24999        0.24999
72     0.00000        0.00000        0.00000
73     0.24984        0.24984        0.24984
74     0.24969        0.24969        0.24969
75     0.00000        0.00000        0.00000

```

# Matlab-Codes

## rbc\_3d\_frei.m

```
1 function rbc_3d_frei
2
3 close all
4
5 %%
6 %-----Variablen-----
7 Xn=[-10.1 -0.98 37.41]';
8 num_dim=3;
9
10 global b r sig
11 b=8/3;
12 sig=10;
13 r=28;
14
15 dt=0.001;
16 t_end=100;
17 %t=0:dt:t_end; %Zeitvektor
18
19 %Tangentenpropergator und Aequivalente Matrix
20 A=eye(num_dim);
21 M=eye(num_dim);
22
23 %Lyapunov-Analyse
24 Lyapunov=zeros(num_dim,t_end/dt);
25 Lyapunov_spec=zeros(num_dim,1);
26
27
28 %%
29 %-----Berechnung-----
30
```

```

31 for i=1:(t_end/dt);
32
33     %Heun Verfahren (nicht linear)
34     Xn_star=Xn(:,i)+dt.*forcing(0,Xn(:,i));
35     Xn(:,i+1)=0.5.*Xn(:,i)+0.5*(Xn_star+dt*forcing(0,Xn_star));
36
37     %Linearer Tangentenoperator:
38     M(:,:)=0.5*(eye(num_dim)+(eye(num_dim)+dt*jacobi(Xn_star))...
39         *(eye(num_dim)+dt*jacobi(Xn(:,i)))));
40
41     %Lyapunov-Analyse:
42     jac=jacobi(Xn(:,i));
43
44     for m=1:num_dim
45         Lyapunov(m,i)=log(norm(jac(m,:)));
46     end
47
48     %Aequivalente Matrix zum Tangentenpropergator
49     A=M*A;
50
51 end
52
53 for m=1:num_dim
54     Lyapunov_spec(m,1)=sum(Lyapunov(m,:))/i;
55 end
56
57 fprintf('Lyapunov Spektrum:\n');
58 fprintf('%f\n',Lyapunov_spec);
59
60
61 %% Plots
62 %Lorenz 1963 Attraktor 3dim
63 fig1=figure('Units','normalized','Position',[0 0 1 1]);
64 set(fig1,'name','Lorenz 1963 Attraktor','numbertitle','off')
65
66 %colordef black
67 plot3(Xn(1,:),Xn(2,:),Xn(3,:), 'r')
68 title('Lorenz 1963 Attraktor');
69 grid on;
70 xlabel('X');

```

```

71 ylabel('Y');
72 zlabel('Z');
73 view([111 -36])
74 grid off
75
76
77 %Plot der Phasenräume in Ebenen
78 fig2=figure('Units','normalized','Position',[0.1 0 0.5 1]);
79 set(fig2,'name','Lorenz 1963 Attraktor','numbertitle','off')
80 subplot(3,1,1)
81 plot(Xn(1,:),Xn(2:,:), 'g')
82 grid on;
83 title('x-y Ebene');
84 xlabel('x');
85 ylabel('y');
86
87 subplot(3,1,2)
88 hold on;
89 plot(Xn(2,:),Xn(3:,:), 'g')
90 grid on;
91 title('y-z Ebene');
92 xlabel('z')
93 ylabel('y')
94
95 subplot(3,1,3)
96 hold on;
97 plot(Xn(1,:),Xn(3:,:), 'g')
98 grid on;
99 title('x-z Ebene');
100 xlabel('x')
101 ylabel('z')
102
103
104 function forcing=forcing(~,X)
105
106     forcing=[-sig sig 0;r -1 -X(1);X(2) 0 -b]*X;
107
108     %forcing=[0 sig 0;r 0 -X(1);X(2) 0 -0]*X; %konservativ
109 end
110

```

```
111 function jacobi=jacobi(X)
112
113     jacobi=[0 sig 0; -X(3)+r 0 -X(1);X(2) X(1) 0];
114 end
115 end
```

**rbc\_3d\_stoerungsdivergenz.m**

```

1 function rbc_3d_stoerungsdivergenz
2
3 close all
4
5 %%
6 %-----Variablen-----
7 Xn=[-10.1 -0.98 37.41]';
8 num_dim=3;
9
10 global b r sig
11 b=8/3;
12 sig=10;
13 r=28;
14
15 dt=0.005;
16 t_end=100;
17 n=500; %Anzahl der Ensembles
18 dt_pert=[0.1 3 5 10];
19
20
21 %Tangentenpropagator und Aequivalente Matrix
22 A=eye(num_dim);
23 M=eye(num_dim);
24 E=eye(num_dim);
25
26 %Beobachtungen
27 %jeweils bis höchstens 3.5% Abweichung vom vorhergesagten Wert
28 xo=zeros(num_dim,1);
29 for_xo=[0.7 0.6 0.2];
30 xo(1,:)=-for_xo(1,1)+for_xo(1,1)*2.*randn(1);
31 xo(2,:)=-for_xo(1,2)+for_xo(1,2)*2.*randn(1);
32 xo(3,:)=-for_xo(1,3)+for_xo(1,3)*2.*randn(1);
33
34 %Fehlerkovarianzmatrizen
35 R=diag(for_xo.^2,0);
36 Q=diag(for_xo.^2.*1.2,0);
37 P=R;
38
39 %Stoerungen

```

```

40 v=zeros(num_dim,n); %zur Stoerungsgenerierung
41 x_pert=zeros(num_dim,n); %Stoerungen
42 x_pert_end=zeros(num_dim,n,4); %integrierte Stoerungen
43
44 %%
45 %-----Berechnung-----
46
47 %Freie Vorhersage zur Generierung der Stoerungen
48 oo=odeset('RelTol',1e-7);
49 sol=ode45(@forcing,[0 t_end],Xn(:,1),oo);
50 odeget(oo,'RelTol');
51 Xnr=deval(sol,0:dt:t_end);
52
53 for i=1:(t_end/dt);
54     %Heun Verfahren (nicht linear)
55     Xn_star=Xn(:,i)+dt.*forcing(0,Xn(:,i));
56     Xn(:,i+1)=0.5.*Xn(:,i)+0.5*(Xn_star+dt*forcing(0,Xn_star));
57
58     %Linearer Tangentenoperator:
59     M(:,:)=0.5*(eye(num_dim)+(eye(num_dim)+dt*jacobi(Xn_star))...
60         *(eye(num_dim)+dt*jacobi(Xn(:,i))));
61
62     %Aequivalente Matrix zum Tangentenpropergator
63     A=M*A;
64
65     %Analyse
66     if i==1
67         x_obs=Xnr(:,i+1)+xo; %Beobachtung
68
69         %Kalman-Filter
70         x=Xn(:,i+1);
71         P=(E-A)'*P*(E-A)+A'*R*A+Q;
72         S=P+R;
73         K=P/S;
74         x=x+K*(x_obs-x);
75
76         P=P-K*P;
77         trace_P_after_update=trace(P);
78
79         %Updateschritt

```



```

80     Xn(:,i+1)=x;
81
82     %Berechnung der singulaeren Vektoren
83     [~,~,V]=svd(A);
84
85     %Erstellung der Stoerungen
86     sigma=sqrt(trace_P_after_update);
87     Alpha=random('norm',0,sigma,1,n);
88     emp_fac=0.3;
89     for j=1:n;
90         if Alpha(1,j)>3*sigma
91             Alpha(1,j)=3*sigma;
92         end
93         if Alpha(1,j)<-3*sigma
94             Alpha(1,j)=-3*sigma;
95         end
96         v(:,j)=emp_fac.*Alpha(1,j).*V(:,1);
97         x_pert(:,j)=v(:,j)+x(:);
98
99         for h=1:length(dt_pert)
100             %Runge-Kutta Verfahren für Störungen:
101             sol=ode45(@forcing,[0 t_end],x_pert(:,j,1));
102             x_pert_end(:,j,h)=deval(sol,dt_pert(h));
103         end
104     end
105 end
106 A=eye(num_dim);    %Aequivalente Matrix zurücksetzen
107 end
108
109
110 %-----Plots-----
111
112 %Lorenz 1963 Attraktor mit divergenten Störungen
113 fig1=figure('Units','normalized','Position',[0 0 1 1]);
114 set(fig1,'name','Lorenz 1963 Attraktor','numbertitle','off')
115
116 for h=1:4
117     subplot(2,2,h)
118     hold on
119     plot3(Xn(1,:),Xn(2,:),Xn(3,:), 'r')

```

```

120     for j=1:n
121         plot3(x_pert_end(1,j,h),x_pert_end(2,j,h),x_pert_end(3,j,h),'b*','
            MarkerSize',10)
122     end
123     title(sprintf('Lorenz 1963 Attraktor:\n t=%g',dt_pert(h)))
124     grid on;
125     xlabel('X');
126     ylabel('y');
127     zlabel('Z');
128     view([49 384])
129 end
130
131
132 %-----Functions-----
133
134 function forcing=forcing(~,X)
135
136     forcing=[-sig sig 0;r -1 -X(1);X(2) 0 -b]*X;
137
138 end
139
140
141 function jacobi=jacobi(X)
142
143     jacobi=[0 sig 0; -X(3)+r 0 -X(1);X(2) X(1) 0];
144 end
145
146 %close all
147
148 end

```

**rbc\_3d\_svd.m**

```

1 function rbc_3d_svd
2
3 close all
4 %%
5 %-----Variablen-----
6 Xn=[-10.1 -0.98 37.41]';
7 num_dim=3;
8
9 global b r sig
10 b=8/3;
11 sig=10;
12 r=28;
13
14 dt=0.001;
15 t_end=100;
16 %t=0:dt:t_end; %Zeitvektor
17
18 %Tangentenpropergator und Aequivalente Matrix
19 A=eye(num_dim);
20 M=eye(num_dim);
21
22 S=zeros(t_end/dt,1);
23 S(1,1)=1;
24
25 %%
26 %-----Berechnung-----
27
28 for i=1:(t_end/dt);
29
30     %Heun Verfahren (nicht linear)
31     Xn_star=Xn(:,i)+dt.*forcing(0,Xn(:,i));
32     Xn(:,i+1)=0.5.*Xn(:,i)+0.5*(Xn_star+dt*forcing(0,Xn_star));
33
34     %Linearer Tangentenoperator:
35     M(:,:)=0.5*(eye(num_dim)+(eye(num_dim)+dt*jacobi(Xn_star))...
36         *(eye(num_dim)+dt*jacobi(Xn(:,i))));
37
38
39     %Aequivalente Matrix zum Tangentenpropergator

```

```

40     A=M*A;
41
42     %Berechnung der Singulaerwerte
43     [~,s,~]=svd(A);
44
45     %if s(1,1)<5
46     S(i+1)=s(1,1);
47     %end
48
49     A=eye(num_dim);    %Aequivalente Matrix zuruecksetzen
50
51 end
52
53
54
55
56 %% Plots
57 %Singulaerwerte der Stoerungsentwicklung am Lorenzattraktor
58 figure
59 hold on
60 for i=2:(t_end/dt);
61     if (S(i)>=1) && (S(i)<1.003)
62         plot3(Xn(1,i),Xn(2,i),Xn(3,i),'y')
63     end
64     if (S(i)>=1.003) && (S(i)<1.006)
65         plot3(Xn(1,i),Xn(2,i),Xn(3,i),'m')
66     end
67     if (S(i)>=1.006) && (S(i)<1.009)
68         plot3(Xn(1,i),Xn(2,i),Xn(3,i),'r')
69     end
70     if (S(i)>=1.009) && (S(i)<1.012)
71         plot3(Xn(1,i),Xn(2,i),Xn(3,i),'g')
72     end
73     if (S(i)>=1.012) && (S(i)<1.015)
74         plot3(Xn(1,i),Xn(2,i),Xn(3,i),'c')
75     end
76     if (S(i)>=1.015) && (S(i)<1.018)
77         plot3(Xn(1,i),Xn(2,i),Xn(3,i),'b')
78     end
79

```

```
80 end
81 title('Singulärwerte der Störungsentwicklung am Lorenzattraktor')
82 grid on;
83 xlabel('X');
84 ylabel('y');
85 zlabel('Z');
86 view([49 384])
87
88
89 function forcing=forcing(~,X)
90
91     forcing=[-sig sig 0;r -1 -X(1);X(2) 0 -b]*X;
92
93     %forcing=[0 sig 0;r 0 -X(1);X(2) 0 -0]*X; %konservativ
94 end
95
96 function jacobi=jacobi(X)
97
98     jacobi=[0 sig 0; -X(3)+r 0 -X(1);X(2) X(1) 0];
99 end
100
101 end
```

**rbc\_3d\_kons\_diss.m**

```

1 function rbc_3d_kons_diss
2
3 close all
4 %%
5 %-----Variablen-----
6 Xn=[-10.1 -0.98 37.41]';
7 Xn_cons=[-10.1 -0.98 37.41]';
8
9 global b r sig
10 b=8/3;
11 sig=10;
12 r=28;
13
14 dt=0.005;
15 t_end=50;
16
17
18
19 %%
20 %-----Berechnung-----
21
22 for i=1:(t_end/dt);
23
24     %Heun Verfahren (nicht linear)
25     Xn_star=Xn(:,i)+dt.*forcing(0,Xn(:,i));
26     Xn(:,i+1)=0.5.*Xn(:,i)+0.5*(Xn_star+dt*forcing(0,Xn_star));
27
28     %Heun Verfahren konservativ (nicht linear)
29     Xn_star_cons=Xn_cons(:,i)+dt.*forcing_cons(0,Xn_cons(:,i));
30     Xn_cons(:,i+1)=0.5.*Xn_cons(:,i)+0.5*(Xn_star_cons+dt*forcing_cons(0,
        Xn_star_cons));
31
32 end
33
34
35 %% Plots
36 %Lorenz 1963 Attraktor
37 fig=figure('Units','centimeters','Position',[0 0 15 20]);
38 set(fig,'name','Lorenz 1963 Attraktor','numbertitle','off')

```

```

39
40 %colordef black
41
42 subplot(2,1,1)
43 plot3(Xn_cons(1,:),Xn_cons(2,:),Xn_cons(3,:), 'r')
44 title('Lorenz 1963 Attraktor');
45 grid on;
46 xlabel('X');
47 ylabel('Y');
48 zlabel('Z');
49 view([49 384])
50 grid off
51 title('Lorenz 1963 Attraktor - konservativ')
52
53 subplot(2,1,2)
54 plot3(Xn(1,:),Xn(2,:),Xn(3,:), 'r')
55 title('Lorenz 1963 Attraktor');
56 grid on;
57 xlabel('X');
58 ylabel('Y');
59 zlabel('Z');
60 view([49 384])
61 grid off
62 title('Lorenz 1963 Attraktor - dissipativ')
63
64
65
66 function forcing=forcing(~,X)
67
68     forcing=[-sig sig 0;r -1 -X(1);X(2) 0 -b]*X;
69
70 end
71
72 function forcing_cons=forcing_cons(~,X)
73
74     forcing_cons=[0 sig 0;r 0 -X(1);X(2) 0 -0]*X; %konservativ
75 end
76
77
78 end

```

**rbc\_3d\_rinne\_roehre.m**

```

1 function rbc_3d_rinne_roehre
2
3 Xn=[-10.1,-0.98,37.41];
4
5 global b r sig
6 b=8/3;
7 sig=10;
8 r=28;
9
10 options = odeset('reltol',1e-6);
11 [~,yy] = ode45(@forcing,[0 20],Xn,options);
12
13 n=20;
14 x=linspace(-20,20,n);
15 y=linspace(-25,25,n);
16 z=linspace(10,50,n);
17
18 [X,Y,Z] = meshgrid(x,y,z);
19
20 V=0.5*(Y.^2+Z.^2)-r*Z;
21 E=0.5*X.^2-sig*Z;
22
23 %close all
24 figure
25 %colordef black
26 hold on
27 p = patch(isosurface(X,Y,Z,V,-347.2458));
28 isonormals(X,Y,Z,V,p)
29 set(p,'FaceColor','g','EdgeColor','none');
30 set(p,'LineWidth',2)
31
32 p = patch(isosurface(X,Y,Z,E,-323.0950));
33 isonormals(X,Y,Z,E,p)
34 set(p,'FaceColor','b','EdgeColor','none');
35 set(p,'LineWidth',2)
36
37 plot3(yy(:,1),yy(:,2),yy(:,3),'Color','r','LineWidth',3);
38
39 alpha(0.95)

```



```
40 view(3); axis tight
41 camlight
42 lighting gouraud
43 light('Position',[1 -1 0],'Style','infinite');
44
45 title('Konservativer Lorenz 1963 Attraktor')
46 xlabel('X')
47 ylabel('Y')
48 zlabel('Z')
49 view(-152,22)
50
51 function forcing = forcing(~,X)
52
53 sig=10;
54 r=28;
55     forcing=[0 sig 0;r 0 -X(1);X(2) 0 0]*X;
56
57 end
58
59 end
```

**rbc\_3d\_normen.m**

```

1 function rbc_3d_normen
2
3 close all
4
5 %%
6 %-----Variablen-----
7 Xn=[-10.1 -0.98 37.41]';
8 num_dim=3;
9
10 global b r sig
11 b=8/3;
12 sig=10;
13 r=28;
14
15 dt=0.001;
16 t_end=30;
17 %t=0:dt:t_end; %Zeitvektor
18 dt_ana=0.1;
19
20 %Tangentenproperator und Aequivalente Matrix
21 A=eye(num_dim);
22 M=eye(num_dim);
23
24
25 %%
26 %-----Berechnung-----
27
28 for i=1:(t_end/dt);
29
30     %Heun Verfahren (nicht linear)
31     Xn_star=Xn(:,i)+dt.*forcing(0,Xn(:,i));
32     Xn(:,i+1)=0.5.*Xn(:,i)+0.5*(Xn_star+dt*forcing(0,Xn_star));
33
34     %Linearer Tangentenoperator:
35     M(:,:)=0.5*(eye(num_dim)+(eye(num_dim)+dt*jacobi(Xn_star))...
36         *(eye(num_dim)+dt*jacobi(Xn(:,i))));
37
38     %Aequivalente Matrix zum Tangentenproperator
39     A=M*A;

```

```

40
41     if mod((i),(dt_ana/dt))==0
42
43         if i==dt_ana/dt
44             X=Xn(:,i+1);
45
46             [~,~,x_pert1]=svd(A);
47             x_pert1=X+5.*x_pert1(:,1);
48             x_pert1_end=A*x_pert1;
49
50             [~,~,x_pert2]=svd(H_norm(X)*A);
51             x_pert2=X+5.*x_pert2(:,1);
52             x_pert2_end=A*x_pert2; %H_norm(X)
53         end
54         A=eye(num_dim);
55     end
56 end
57
58
59 %% Plots
60 %Lorenz 1963 Attraktor
61 fig1=figure('Units','normalized','Position',[0 0 1 1]);
62 set(fig1,'name','Lorenz 1963 Attraktor','numbertitle','off')
63 %set(fig9,'Color',[0 0 0])
64
65 %colordef black
66 hold on
67 plot3(Xn(1,:),Xn(2,:),Xn(3,:), 'r')
68 plot3(X(1,:),X(2,:),X(3,:), 'k.', 'MarkerSize',25)
69
70 plot3(x_pert1(1),x_pert1(2),x_pert1(3), 'g.', 'MarkerSize',25)
71 plot3(x_pert2(1),x_pert2(2),x_pert2(3), 'b.', 'MarkerSize',25)
72
73 plot3(x_pert1_end(1),x_pert1_end(2),x_pert1_end(3), 'g.', 'MarkerSize',25)
74 plot3(x_pert2_end(1),x_pert2_end(2),x_pert2_end(3), 'b.', 'MarkerSize',25)
75
76 line([x_pert1(1),x_pert1_end(1)], [x_pert1(2),x_pert1_end(2)],...
77      [x_pert1(3),x_pert1_end(3)], 'Color','g')
78 line([x_pert2(1),x_pert2_end(1)], [x_pert2(2),x_pert2_end(2)]...
79      , [x_pert2(3),x_pert2_end(3)], 'Color','b')

```

```

80
81 view([-317 36])
82 title('Lorenz 1963 Attraktor mit Störungen');
83 grid on;
84 xlabel('X');
85 ylabel('Y');
86 zlabel('Z');
87 legend('Attraktor','Zustand','Eulerstörung','Energiestörung',2)
88
89 %%
90 %Functions
91
92 function forcing=forcing(~,X)
93
94     forcing=[-sig sig 0;r -1 -X(1);X(2) 0 -b]*X;
95
96     %forcing=[0 sig 0;r 0 -X(1);X(2) 0 0]*X; %konservativ
97 end
98
99 function jacobi=jacobi(X)
100
101     jacobi=[0 sig 0; -X(3)+r 0 -X(1);X(2) X(1) 0];
102 end
103
104 function H_norm=H_norm(~)
105     H_norm=zeros(num_dim);
106     H_norm(2,2)=sqrt(1/2);
107     H_norm(3,3)=sqrt(1/2);
108
109 end
110
111
112 end

```

**rbc\_6d\_imagcheck.m**

```

1 function rbc_6d_imagcheck
2 %%
3 %-----Modellparameter-----
4
5 Xn=[1 1 1 1 1 1]';
6 num_dim=length(Xn);
7
8 dt=0.001;
9 t_end=10;
10 t=0:dt:t_end; %Zeitvektor
11
12 global a b c e f r sig
13 a=1; b=1; c=1/(2*b); e=a^3/(a^2*(1+a^2)); f=2*a^3/(a^2*b^2*(1+a^2)^2);
14 sig=10;
15 r=28;
16
17 %Eintraege für function: forcing
18 varf_1=a*(a^2-3)*pi^2/(2*b*(1+a^2));
19 varf_2=a*e*r*sig/(b*pi*(1+a^2));
20 %varf_3=wird in function definiert, da abh. von akt. Zustand
21 varf_4=a*pi^2/(2*b);
22 varf_5=a*b*e*pi^2/(2*f);
23
24 u=0;
25
26 H_norm_Mat_check=zeros(1,t_end/dt+1);
27 C_norm_Mat_check=zeros(1,t_end/dt+1);
28 S_norm_Mat_check=zeros(1,t_end/dt+1);
29
30 %Freie Vorhersage zur Generierung der Beobachtungen
31 oo=odeset('RelTol',1e-7);
32 sol=ode45(@forcing,[0 t_end],Xn(:,1),oo);
33 odeget(oo,'RelTol');
34 Xnr=deval(sol,0:dt:t_end); %freie Vorhersage
35
36 %Test ob Matrizen der Normen richtig sind bzw. ob sie imaginär werden
37 for i=1:t_end/dt+1
38     H_norm_Mat_check(1,i)=(H_norm(Xnr(:,i))*Xnr(:,i))'*(H_norm(Xnr(:,i))*Xnr(:,
        i));

```

```

39     C_norm_Mat_check(1,i)=(C_norm(Xnr(:,i))*Xnr(:,i))'*(C_norm(Xnr(:,i))*Xnr(:,
        i));
40     S_norm_Mat_check(1,i)=(S_norm(Xnr(:,i))*Xnr(:,i))'*(S_norm(Xnr(:,i))*Xnr(:,
        i));
41
42     H_imag_norm=norm(imag(H_norm(Xnr(:,i))));
43     C_imag_norm=norm(imag(C_norm(Xnr(:,i))));
44     S_imag_norm=norm(imag(S_norm(Xnr(:,i))));
45
46     if H_imag_norm~=0
47         disp('H')
48         disp(H_imag_norm)
49         u=u+1;
50     end
51
52     if C_imag_norm~=0
53         disp('C')
54         disp(C_imag_norm)
55         u=u+1;
56     end
57
58     if S_imag_norm~=0
59         disp('S')
60         disp(S_imag_norm)
61         u=u+1;
62     end
63
64 end
65 %colordef black
66 figure('Units','normalized','Position',[0 0 1 1]);
67
68 subplot(3,1,1)
69 plot(t,H_norm_Mat_check(1,:), 'k')
70 title('Zeitlicher Verlauf Hamiltonfunktion, berechnet mit Norm-Matrix')
71 xlabel('Zeit')
72 ylabel('Energienorm')
73
74 subplot(3,1,2)
75 plot(C_norm_Mat_check(1,:), 'k')
76 title('Zeitlicher Verlauf Helizitäts-Casimir, berechnet mit Norm-Matrix')

```

```

77 xlabel('Zeit')
78 ylabel('Helizitätsnorm')
79
80 subplot(3,1,3)
81 plot(S_norm_Mat_check(1,:), 'k')
82 title('Zeitlicher Verlauf Temperatur-Casimir, berechnet mit Norm-Matrix')
83 xlabel('Zeit')
84 ylabel('Temperaturnorm')
85
86
87
88 %-----Functions-----
89
90 function forcing=forcing(~,X)
91
92     varf_3=a*pi*(b*f*pi*X(6)+b)/e;
93
94     forcing=[0 varf_1*X(3) 0 0 varf_2 0; ...
95             -varf_1*X(3) 0 0 -varf_2 0 0; ...
96             0 0 0 0 0 0; ...
97             0 -varf_3 varf_4*X(5) 0 0 0; ...
98             varf_3 0 -varf_4*X(4) 0 0 0;...
99             -varf_5*X(5) varf_5*X(4) 0 0 0 0]*X;
100
101 end
102
103 function H_norm=H_norm(X)
104     H_norm=zeros(num_dim);
105     H_norm(1,1)=b*pi*sqrt((1+a^2)/4*a);
106     H_norm(2,2)=H_norm(1,1);
107     H_norm(3,3)=pi/b*sqrt(1/(2*a));
108     H_norm(6,6)=sqrt(r*f*sig/(a*pi*X(6)));
109 end
110
111 % function H_norm=H_norm(X)
112 %     H_norm=zeros(num_dim);
113 %     H_norm(1,1)=sqrt(1/(4*a*b^2*pi*X(1)^2)*((1+a^2)*b^4*pi^3*(X(1)^2+X(2)^2)
114 %         ...
115 %         +2*pi^3*X(3)^2+4*r*b^2*f*sig*X(6)));
116 %

```

```

116 % end
117
118 function C_norm=C_norm(X) %nicht quadratisch
119     C_norm=zeros(num_dim);
120     varx=(1+a)^2*b^2*e*pi;
121     C_norm(1,1)=sqrt(varx*X(4)/X(1));
122     C_norm(2,2)=sqrt(varx*X(5)/X(2));
123     C_norm(3,3)=sqrt(4*f*pi*X(6)/X(3)+4/X(3));
124     C_norm=sqrt(pi/(2*a*b))*C_norm;
125 end
126
127
128 % function C_norm=C_norm(X) %C, alles in erste Komponente gestopft
129 %     C_norm=zeros(num_dim);
130 %     C_norm(1,1)=sqrt((pi/(2*a*b*X(1)^2)*((1+a^2)*b^2*e*pi*(X(1)*X(4)+X(2)
131 %         *X(5))...
132 %         +4*f*pi*X(3)*X(6)+4*X(3))));
133 % end
134
135 function S_norm=S_norm(X)
136     S_norm=zeros(num_dim);
137     S_norm(4,4)=sqrt(r*sig*e^2/(4*a));
138     S_norm(5,5)=S_norm(4,4);
139     S_norm(6,6)=sqrt(r*sig/(12*a*pi)*(6*f^2*pi+12*f/X(6)));
140 end
141
142 % function S_norm=S_norm(X) %S, alles in erste Komponente gestopft
143 %     S_norm=zeros(num_dim);
144 %     S_norm(1,1)=sqrt(r*sig/(12*a*pi*X(1)^2)*(3*e^2*pi*(X(4)^2+X(5)^2)...
145 %         +6*f^2*pi*X(6)^2+12*f*X(6)));
146 % end
147 end

```



**rbc\_6d\_phasenraum.m**

```

1 function rbc_6d_phasenraum
2
3 Xn=[1 1 0 1 1 1]';
4
5 global a b c e f r sig
6 a=1; b=1; c=1/(2*b); e=a^3/(a^2*(1+a^2)); f=2*a^3/(a^2*b^2*(1+a^2)^2);
7 sig=10;
8 r=28;
9
10 t_end=50;
11 dt=0.001;
12
13 %Eintraege für function
14 varf_1=a*(a^2-3)*pi^2/(2*b*(1+a^2));
15 varf_2=a*e*r*sig/(b*pi*(1+a^2));
16 %varf_3=wird in function definiert
17 varf_4=a*pi^2/(2*b);
18 varf_5=a*b*e*pi^2/(2*f);
19
20 %Freie Vorhersage
21 oo=odeset('RelTol',1e-7);
22 sol=ode45(@forcing,[0 t_end],Xn(:,1),oo);
23 odeget(oo,'RelTol');
24 Xnr=deval(sol,0:dt:t_end); %freie Vorhersage
25
26 %%
27 %-----Plots-----
28 delete out.ps
29 for_axes=['A' 'B' 'C' 'D' 'E' 'F'];
30
31 %Phasenraumplots 2d
32 fig=figure('Units','normalized','Position',[0 0 1 1]);
33 set(fig,'name','Phasenraumdiagramme 2d','numbertitle','off')
34 l=1;
35 for xx=1:6
36     for yy=1:xx-1
37         subplot(4,3,l)
38         if xx~=3 && yy~=3
39             plot(Xnr(xx,:),Xnr(yy,:))

```

```

40         xlabel(for_axes(xx))
41         ylabel(for_axes(yy))
42         title(sprintf('Komponenten %s und %s',for_axes(yy),for_axes(xx)))
43         l=l+1;
44     end
45 end
46 end
47 print(gcf,'-dpasc','-append','out.ps')
48 print(gcf, '-depasc', 'ergebnisse/2d_allen.eps');
49 close(gcf)
50
51 %Phasenraumplots 3d
52 fig=figure('Units','normalized','Position',[0 0 1 1]);
53 set(fig,'name','Phasenraumdiagramme 3d','numbertitle','off')
54 l=1;
55 for xx=1:6
56     for yy=1:xx-1
57         for zz=1:yy-1
58             subplot(4,3,1)
59             if xx~=3 && yy~=3 && zz~=3
60                 plot3(Xnr(xx,:),Xnr(yy,:),Xnr(zz,:))
61                 grid on
62                 xlabel(for_axes(xx))
63                 ylabel(for_axes(yy))
64                 zlabel(for_axes(zz))
65                 title(sprintf('Komponenten %s, %s und %s',for_axes(zz),for_axes(
66                     yy),for_axes(xx)))
67                 l=l+1;
68             end
69         end
70     end
71 end
72 print(gcf,'-dpasc','-append','out.ps')
73 print(gcf, '-depasc', 'ergebnisse/3d_allen.eps');
74 close(gcf)
75
76 %Phasenraumplots 2d
77 l=1;
78 for xx=1:6
79     for yy=1:xx-1

```

```

79     if xx~=3 && yy~=3
80         fig=figure('Units','normalized','Position',[0 0 1 1]);
81         set(fig,'name','Phasenraumdiagramm 2d','numbertitle','off')
82         plot(Xnr(xx,:),Xnr(yy,:))
83         xlabel(for_axes(xx))
84         ylabel(for_axes(yy))
85         title(sprintf('Phasenraum: Komponenten %s und %s',for_axes(yy),
86                     for_axes(xx)))
87         l=l+1;
88         eps_name=sprintf('ergebnisse/2d_Komp_%s_%s.eps',for_axes(xx),
89                     for_axes(yy));
90         print(gcf, '-depsc', eps_name);
91         print(gcf, '-dpsc', '-append', 'out.ps')
92         close(gcf)
93     end
94 end
95
96 %Phasenraumplots 3d
97
98 l=1;
99 for xx=1:6
100     for yy=1:xx-1
101         for zz=1:yy-1
102             if xx~=3 && yy~=3 && zz~=3
103                 fig=figure('Units','normalized','Position',[0 0 1 1]);
104                 set(fig,'name','Phasenraumdiagramme 3d','numbertitle','off')
105                 plot3(Xnr(xx,:),Xnr(yy,:),Xnr(zz,:))
106                 grid on
107                 xlabel(for_axes(xx))
108                 ylabel(for_axes(yy))
109                 zlabel(for_axes(zz))
110                 title(sprintf('Phasenraum: Komponenten %s, %s und %s',...
111                             for_axes(zz),for_axes(yy),for_axes(xx)))
112                 l=l+1;
113                 eps_name=sprintf('ergebnisse/3d_Komp_%s_%s_%s.eps'...
114                             ,for_axes(xx),for_axes(yy),for_axes(zz));
115                 print(gcf, '-depsc', eps_name);
116                 print(gcf, '-dpsc', '-append', 'out.ps')

```

```

117         close(gcf)
118     end
119 end
120 end
121 end
122
123
124 %post script file in pdf umwandeln
125 pdf_name=sprintf('phasenraum_for_t_end_%g.pdf',t_end);
126 ps2pdf('psfile', 'out.ps', 'pdffile', pdf_name, ...
127     'gspapersize', 'a4', 'verbose', 1, ...
128     'gscommand', 'C:\Program Files\gs\gs9.06\bin\gswin32c.exe', ...
129     'gsfontpath', 'C:\Program Files\gs\gs9.06\lib', ...
130     'gslibpath', 'C:\Program Files\gs\gs9.06\lib')
131
132
133 function forcing=forcing(~,X)
134
135     varf_3=a*pi*(b*f*pi*X(6)+b)/e;
136
137     forcing=[0 varf_1*X(3) 0 0 varf_2 0; ...
138         -varf_1*X(3) 0 0 -varf_2 0 0; ...
139         0 0 0 0 0 0; ...
140         0 -varf_3 varf_4*X(5) 0 0 0; ...
141         varf_3 0 -varf_4*X(4) 0 0 0;...
142         -varf_5*X(5) varf_5*X(4) 0 0 0 0]*X;
143
144 end
145
146 end

```

**rbc\_6d\_frei.m**

```

1 function rbc_6d_frei
2
3 close all
4
5 Xn=[1 1 1 1 1 1]';
6 num_dim=6;
7
8 global a b c e f r sig
9 a=1; b=1; c=1/(2*b); e=a^3/(a^2*(1+a^2)); f=2*a^3/(a^2*b^2*(1+a^2)^2);
10 sig=10;
11 r=28;
12
13 dt=0.001;
14 t_end=100;
15 t=0:dt:t_end;
16
17 var_1=a*(a^2-3)*pi^2/(2*b*(1+a^2));
18 var_2=a*e*r*sig/(b*pi*(1+a^2));
19 %var_3=a*pi*(b*f*pi*X(6)+b)/e;
20 var_4=a*pi^2/(2*b);
21 var_5=a*b*e*pi^2/(2*f);
22
23 Lyapunov=zeros(num_dim,t_end/dt);
24 Lyapunov_spec=zeros(num_dim,1);
25
26
27 for i=1:(t_end/dt);
28
29     %Heun Verfahren (nicht linear)
30     Xn_star=Xn(:,i)+dt.*forcing(0,Xn(:,i));
31     Xn(:,i+1)=0.5.*Xn(:,i)+0.5*(Xn_star+dt*forcing(0,Xn_star));
32
33     %Lyapunov-Analyse:
34     jac=jacobi(Xn(:,i));
35
36     for m=1:num_dim
37         Lyapunov(m,i)=log(norm(jac(m,:)));
38     end
39

```

```

40 end
41
42 for m=1:num_dim
43     Lyapunov_spec(m,1)=sum(Lyapunov(m,:))/i;
44 end
45
46 fprintf('Lyapunov Spektrum:\n');
47 fprintf('%f\n',Lyapunov_spec);
48
49
50 %%
51 %-----Plots-----
52
53 for_axes=['A' 'B' 'C' 'D' 'E' 'F'];
54
55 %Entwickelte Komponenten
56 figure('Units','normalized','Position',[0 0 1 1])
57 for m=1:num_dim
58     subplot(3,2,m)
59     hold on
60     plot(t,Xn(m,:));
61     xlim([0 t_end])
62     xlabel('t')
63     ylabel(sprintf('%s',for_axes(m)))
64     title(sprintf('Komponente %s',for_axes(m)))
65 end
66
67 %%
68 %-----Functions-----
69
70 function forcing=forcing(~,X)
71
72 %     var_1=a*(a^2-3)*pi^2/(2*b*(1+a^2));
73 %     var_2=a*e*r*sig/(b*pi*(1+a^2));
74     var_3=a*pi*(b*f*pi*X(6)+b)/e;
75 %     var_4=a*pi^2/(2*b);
76 %     var_5=a*b*e*pi^2/(2*f);
77
78     forcing=[0 var_1*X(3) 0 0 var_2 0; ...
79             -var_1*X(3) 0 0 -var_2 0 0; ...

```

```

80         0 0 0 0 0 0; ...
81         0 -var_3 var_4*X(5) 0 0 0; ...
82         var_3 0 -var_4*X(4) 0 0 0;...
83         -var_5*X(5) var_5*X(4) 0 0 0 0]*X;
84
85     end
86
87     function jacobi=jacobi(X)
88
89         var_1=a*(a^2-3)*pi^2/(2*b*(1+a^2));
90         var_2=a*e*r*sig/(b*pi*(1+a^2));
91         var_3=a*pi*(b*f*pi*X(6)+b)/e;
92         var_4=a*pi^2/(2*b);
93         var_5=a*b*f*pi^2/e;
94         var_6=a*b*e*pi^2/(2*f);
95
96         jacobi=[0 var_1*X(3) var_1*X(2) 0 var_2 0; ...
97             -var_1*X(3) 0 -var_1*X(1) -var_2 0 0; ...
98             0 0 0 0 0 0; ...
99             0 -var_3 var_4*X(5) 0 var_4*X(3) -var_5*X(2); ... %!!!
100            var_3 0 -var_4*X(4) -var_4*X(3) 0 var_5*X(1); ...
101            -var_6*X(5) var_6*X(4) 0 var_6*X(2) -var_6*X(1) 0];
102
103     end
104
105     %close all
106     end

```

## rbc\_6dim.m

```

1 function [ BS_Mat, BS_rel_Mat, BS_res_Mat ] = rbc_6dim(dt,t_end,dt_ana,n,c1,c2,
    emp_fac,xostd,Alphastd)
2
3 close all
4
5 plot_yn=0;      %(n=0, y=1)
6 stoercheck_yn=0;
7 imagcheck_yn=0;
8
9 %%
10 % Autor: Johanna Bugkel
11 % Datum: 2012
12 % Inhalt:
13 %     Modellparameter
14 %     Variablendefinition
15 %     Berechnung
16 %     Verifikation
17 %     Plots
18 %     Functions
19
20
21 %%
22 %-----Modellparameter-----
23 for Modellparameter=1:1 %Pseudoschleife zur Navigation im Code
24
25 Xn=[1 1 4 1 1 1]'; %Anfangszustand modifiziert
26 %Xn=[1 1 1 1 1 1]'; %Anfangszustand standard
27 num_dim=length(Xn);
28
29 fc_steps=dt_ana;
30 %fc_steps=dt_ana.*[1 5]; %verschiedene Vorhersageintervalle
31 num_fc_steps=length(fc_steps);
32
33 %dt=0.001;
34 %t_end=20;
35 t=0:dt:t_end; %Zeitvektor
36 %dt_ana=0.1;
37 t_ana=dt_ana:dt_ana:t_end; %Zeitvektor Analysen
38

```



```

39 p=0.03;          %prozent. Abweichung der Beob. zur ges. Schwankungsbreite
40                  %entspricht Standardabweichung
41 %n=30;           %Anzahl der Ensembles
42
43 th=0:0.1:1;      %tresholds fuer ROC-Diagramm
44
45 global a b c e f r sig %#ok<TLEV>
46 a=1;
47 b=1.102;         %geometrischer Faktor modifiziert
48 %b=1;
49 c=1/(2*b); e=a^3/(a^2*(1+a^2)); f=2*a^3/(a^2*b^2*(1+a^2)^2);
50 sig=10;
51 r=28;
52
53 num_fc=t_end/dt_ana-(max(fc_steps)-min(fc_steps))/dt_ana-1; %Anzahl der
    Vorhersagen fuer die Verifikation
54
55 %Eintraege fuer function: forcing
56 varf_1=a*(a^2-3)*pi^2/(2*b*(1+a^2));
57 varf_2=a*e*r*sig/(b*pi*(1+a^2));
58 %varf_3=wird in function definiert, da abh. von akt. Zustand
59 varf_4=a*pi^2/(2*b);
60 varf_5=a*b*e*pi^2/(2*f);
61
62 u=0;            %Zähler für Imaginärteile
63
64 end
65
66 %%
67 %-----Variablendefinition-----
68 for Variablendefinition=1:1
69
70 %Zeitliche Entwicklung der Normen
71 E_norm_Xnr=zeros(1,t_end/dt+1);
72 C_norm_Xnr=zeros(1,t_end/dt+1);
73 H_norm_Xnr=zeros(1,t_end/dt+1);
74 S_norm_Xnr=zeros(1,t_end/dt+1);
75
76 H_norm_Mat_check=zeros(1,t_end/dt+1);
77 C_norm_Mat_check=zeros(1,t_end/dt+1);

```

```

78 S_norm_Mat_check=zeros(1,t_end/dt+1);
79
80 %Beobachtungen
81 I=zeros(num_dim,1);
82 I_obs=zeros(num_dim,1);
83 x_obs=zeros(num_dim,t_end/dt_ana);
84 x_obs_yn=zeros(num_dim,num_fc);
85
86 %Tangentenpropagator und Aequivalente Matrix
87 A=eye(num_dim);
88 M=eye(num_dim);
89 E=eye(num_dim);
90
91 %nichtlinear vorhergesagter Zustand
92 x_fc=zeros(num_dim,t_end/dt_ana);
93
94 %Zustand nach Update
95 x=zeros(num_dim,t_end/dt_ana);
96
97 %Spur von P
98 trace_P_before_update=zeros(1,t_end/dt_ana);
99 trace_P_after_update=zeros(1,t_end/dt_ana);
100
101 %Winkel zwischen den singulaeren Vektoren
102 angle_12=zeros(1,t_end/dt_ana-1);
103 angle_13=zeros(1,t_end/dt_ana-1);
104 angle_23=zeros(1,t_end/dt_ana-1);
105
106 %Stoerungen
107 v1=zeros(num_dim,n,t_end/dt_ana); %zur Stoerungsgenerierung
108 v2=zeros(num_dim,n,t_end/dt_ana);
109 v3=zeros(num_dim,n,t_end/dt_ana);
110 x_pert1=zeros(num_dim,n,t_end/dt_ana); %Stoerungen
111 x_pert2=zeros(num_dim,n,t_end/dt_ana);
112 x_pert3=zeros(num_dim,n,t_end/dt_ana);
113 x_pert_end1=zeros(num_dim,n,t_end/dt_ana,num_fc_steps); %integrierte Stoerungen
114 x_pert_end2=zeros(num_dim,n,t_end/dt_ana,num_fc_steps);
115 x_pert_end3=zeros(num_dim,n,t_end/dt_ana,num_fc_steps);
116
117 norm_v1=zeros(1,n*t_end/dt_ana);

```

```

118
119 %Zugeschnittene Zeitreihen
120 to_cut_front=zeros(1,num_fc_steps);
121 to_cut_back=zeros(1,num_fc_steps);
122
123 x_pert_end_cut1=zeros(num_dim,n,num_fc,num_fc_steps);
124 x_pert_end_cut2=zeros(num_dim,n,num_fc,num_fc_steps);
125 x_pert_end_cut3=zeros(num_dim,n,num_fc,num_fc_steps);
126
127 %Vorhergesagte Wahrscheinlichkeiten (aus Ensemble)
128 fc_prob1=zeros(num_dim,num_fc,num_fc_steps);
129 fc_prob2=zeros(num_dim,num_fc,num_fc_steps);
130 fc_prob3=zeros(num_dim,num_fc,num_fc_steps);
131
132 %Brier Score
133 for_BS1=zeros(num_dim,num_fc,num_fc_steps);
134 for_BS2=zeros(num_dim,num_fc,num_fc_steps);
135 for_BS3=zeros(num_dim,num_fc,num_fc_steps);
136 BS1=zeros(num_dim,num_fc_steps);
137 BS2=zeros(num_dim,num_fc_steps);
138 BS3=zeros(num_dim,num_fc_steps);
139 num_pos_fc1=zeros(num_dim,size(th,2),num_fc_steps); %Zahl der positiv
    vorhergesagten Ereignisse
140 num_pos_fc2=zeros(num_dim,size(th,2),num_fc_steps);
141 num_pos_fc3=zeros(num_dim,size(th,2),num_fc_steps);
142 num_pos_fc_for_cond1=zeros(num_dim,size(th,2),num_fc_steps);
143 num_pos_fc_for_cond2=zeros(num_dim,size(th,2),num_fc_steps);
144 num_pos_fc_for_cond3=zeros(num_dim,size(th,2),num_fc_steps);
145 hit1=zeros(num_dim,size(th,2),num_fc_steps); %hits
146 hit2=zeros(num_dim,size(th,2),num_fc_steps);
147 hit3=zeros(num_dim,size(th,2),num_fc_steps);
148 cond_prob1=zeros(num_dim,size(th,2),num_fc_steps); %bedingte
    Wahrscheinlichkeiten
149 cond_prob2=zeros(num_dim,size(th,2),num_fc_steps);
150 cond_prob3=zeros(num_dim,size(th,2),num_fc_steps);
151 clim_prob=zeros(num_dim,1); %klimatologische Wahrscheinlichkeiten
152 for_BS_rel1=zeros(num_dim,size(th,2),num_fc_steps);
153 for_BS_rel2=zeros(num_dim,size(th,2),num_fc_steps);
154 for_BS_rel3=zeros(num_dim,size(th,2),num_fc_steps);
155 for_BS_res1=zeros(num_dim,size(th,2),num_fc_steps);

```

```

156 for_BS_res2=zeros(num_dim,size(th,2),num_fc_steps);
157 for_BS_res3=zeros(num_dim,size(th,2),num_fc_steps);
158 BS_rel1=zeros(num_dim,num_fc_steps); %BS Reliability
159 BS_rel2=zeros(num_dim,num_fc_steps);
160 BS_rel3=zeros(num_dim,num_fc_steps);
161 BS_res1=zeros(num_dim,num_fc_steps); %BS Resolution
162 BS_res2=zeros(num_dim,num_fc_steps);
163 BS_res3=zeros(num_dim,num_fc_steps);
164 BS_unc1=zeros(num_dim,num_fc_steps); %BS Uncertainty
165 BS_Mat=zeros(3,num_dim,num_fc_steps);
166 BS_rel_Mat=zeros(3,num_dim,num_fc_steps);
167 BS_res_Mat=zeros(3,num_dim,num_fc_steps);
168 BS_unc_Mat=zeros(3,num_dim,num_fc_steps);
169
170 %Ensemblespread
171 x_pert_end_mean1=zeros(num_dim,num_fc,num_fc_steps); %Ensemblemittel
172 x_pert_end_mean2=zeros(num_dim,num_fc,num_fc_steps);
173 x_pert_end_mean3=zeros(num_dim,num_fc,num_fc_steps);
174 for_spread1=zeros(num_dim,n,num_fc,num_fc_steps);
175 for_spread2=zeros(num_dim,n,num_fc,num_fc_steps);
176 for_spread3=zeros(num_dim,n,num_fc,num_fc_steps);
177 spread1=zeros(num_dim,num_fc,num_fc_steps); %Spread
178 spread2=zeros(num_dim,num_fc,num_fc_steps);
179 spread3=zeros(num_dim,num_fc,num_fc_steps);
180 spread_norm1=zeros(num_fc,num_fc_steps);
181 % spread_norm2=zeros(num_fc,num_fc_steps);
182 % spread_norm3=zeros(num_fc,num_fc_steps);
183
184 %RMS
185 for_rms=zeros(num_dim,num_fc);
186 rms=zeros(num_dim,1);
187 for_rms_norm=zeros(1,num_fc);
188
189 %ROC-Diagramm
190 Hit1=zeros(num_dim,size(th,2),num_fc_steps);
191 False1=zeros(num_dim,size(th,2),num_fc_steps);
192 Miss1=zeros(num_dim,size(th,2),num_fc_steps);
193 Zero1=zeros(num_dim,size(th,2),num_fc_steps);
194
195 Hit2=zeros(num_dim,size(th,2),num_fc_steps);

```

```

196 False2=zeros(num_dim,size(th,2),num_fc_steps);
197 Miss2=zeros(num_dim,size(th,2),num_fc_steps);
198 Zero2=zeros(num_dim,size(th,2),num_fc_steps);
199
200 Hit3=zeros(num_dim,size(th,2),num_fc_steps);
201 False3=zeros(num_dim,size(th,2),num_fc_steps);
202 Miss3=zeros(num_dim,size(th,2),num_fc_steps);
203 Zero3=zeros(num_dim,size(th,2),num_fc_steps);
204
205 POD1=zeros(num_dim,size(th,2),num_fc_steps); %Probability of detection
206 POFD1=zeros(num_dim,size(th,2),num_fc_steps); %Probability of false detection
207
208 POD2=zeros(num_dim,size(th,2),num_fc_steps);
209 POFD2=zeros(num_dim,size(th,2),num_fc_steps);
210
211 POD3=zeros(num_dim,size(th,2),num_fc_steps);
212 POFD3=zeros(num_dim,size(th,2),num_fc_steps);
213
214 Diff12_01=zeros(num_dim,1);
215 Diff13_01=zeros(num_dim,1);
216
217 %Lyapunov Analyse
218 Lyapunov=zeros(num_dim,t_end/dt_ana);
219 Lyapunov_spec=zeros(num_dim,1);
220
221 end
222
223 %%
224 %-----Berechnung-----
225 for Berechnung=1:1
226
227 %Freie Vorhersage zur Generierung der Beobachtungen
228 oo=odeset('RelTol',1e-7);
229 sol=ode45(@forcing,[0 t_end],Xn(:,1),oo);
230 odeget(oo,'RelTol');
231 Xnr=deval(sol,0:dt:t_end); %freie Vorhersage
232
233 %xo=randn(num_dim,t_end/dt_ana);
234 xo=xostd;
235 for m=1:num_dim

```

```

236     I(m)=range(Xnr(m,:)); %Schwankungsbreite der Komponenten
237     I_obs(m)=p.*I(m);      %Schwankungsbreite fuer Beobachtungen
238     xo(m,:)=xo(m,:)*I_obs(m);
239 end
240
241 %Fehlerkovarianzmatrizen
242 R=diag(I_obs.^2,0);
243 Q=diag(I_obs.^2.*1.2,0);
244 P=diag(I_obs.^2,0);
245 Q(3,3)=0.2;
246
247 trace_P_before_update(1)=trace(P);
248 trace_P_after_update(1)=trace(P);
249
250 %Test, ob Erhaltungsgrößen konstant
251 for i=1:t_end/dt+1
252     E_norm_Xnr(1,i)=norm(Xnr(:,i));
253     H_norm_Xnr(1,i)=H_norm_cons_check(Xnr(:,i));
254     C_norm_Xnr(1,i)=C_norm_cons_check(Xnr(:,i));
255     S_norm_Xnr(1,i)=S_norm_cons_check(Xnr(:,i));
256 end
257
258 %Test ob Matrizen der Normen richtig sind bzw. ob sie imaginär werden
259 if imagcheck_yn==1
260
261     for i=1:t_end/dt+1
262         H_norm_Mat_check(1,i)=(H_norm(Xnr(:,i))*Xnr(:,i))'*(H_norm(Xnr(:,i))*Xnr(:,
            i));
263         C_norm_Mat_check(1,i)=(C_norm(Xnr(:,i))*Xnr(:,i))'*(C_norm(Xnr(:,i))*Xnr(:,
            i));
264         S_norm_Mat_check(1,i)=(S_norm(Xnr(:,i))*Xnr(:,i))'*(S_norm(Xnr(:,i))*Xnr(:,
            i));
265
266         H_imag_norm=norm(imag(H_norm(Xnr(:,i))));
267         C_imag_norm=norm(imag(C_norm(Xnr(:,i))));
268         S_imag_norm=norm(imag(S_norm(Xnr(:,i))));
269
270         if mod((i),(dt_ana/dt))==0
271             if H_imag_norm~=0
272                 disp('H')

```

```

273         disp(H_imag_norm)
274         u=u+1;
275     end
276
277     if C_imag_norm~=0
278         disp('C')
279         disp(C_imag_norm)
280         u=u+1;
281     end
282
283     if S_imag_norm~=0
284         disp('S')
285         disp(S_imag_norm)
286         u=u+1;
287     end
288 end
289
290 end
291
292 %colordef black
293 figure('Units','normalized','Position',[0 0 1 1]);
294
295 subplot(3,1,1)
296 plot(t,H_norm_Mat_check(1,:), 'g')
297 title('Zeitlicher Verlauf Hamiltonfunktion, berechnet mit Norm-Matrix')
298 xlabel('Zeit')
299 ylabel('Energienorm')
300
301 subplot(3,1,2)
302 plot(C_norm_Mat_check(1,:), 'g')
303 title('Zeitlicher Verlauf Helizitäts-Casimir, berechnet mit Norm-Matrix')
304 xlabel('Zeit')
305 ylabel('Helizitätsnorm')
306
307 subplot(3,1,3)
308 plot(S_norm_Mat_check(1,:), 'g')
309 title('Zeitlicher Verlauf Temperatur-Casimir, berechnet mit Norm-Matrix')
310 xlabel('Zeit')
311 ylabel('Temperaturnorm')
312

```

```

313 end
314
315 %EPS
316 k=0;
317 for i=1:(t_end/dt);
318     %Heun Verfahren (nicht linear)
319     Xn_star=Xn(:,i)+dt.*forcing(0,Xn(:,i));
320     Xn(:,i+1)=0.5.*Xn(:,i)+0.5*(Xn_star+dt*forcing(0,Xn_star));
321
322     %Linearer Tangentenoperator:
323     M(:,:)=0.5*(eye(num_dim)+(eye(num_dim)+dt*jacobi(Xn_star))...
324         *(eye(num_dim)+dt*jacobi(Xn(:,i))));
325
326     %Lyapunov-Analyse:
327     jac=jacobi(Xn(:,i));
328
329     for m=1:num_dim
330         Lyapunov(m,i)=log(norm(jac(m,:)));
331     end
332
333     %Aequivalente Matrix zum Tangentenpropagator
334     A=M*A;
335
336     %Analyse
337     if mod((i),(dt_ana/dt))==0 %z.B. i=200, t=0.2 für dt_ana/dt=200
338
339         %Test ob lin. Störungswachstum mit nichtlin. uebereinstimmt
340         if stoercheck_yn==1
341             if k>0
342                 B=A;
343                 B(3,:)=[];
344                 B(:,3)=[];
345                 [~,~,v2test]=svd(B);
346                 va=[v2test(1:2,1); 0; v2test(3:5,1)];
347
348                 Xs=x(:,k)+0.5*va;
349
350                 sol=ode45(@forcing,[0 dt_ana],Xs,oo);
351                 Xsfc=deval(sol,0:dt:dt_ana);
352                 sol=ode45(@forcing,[0 dt_ana],x(:,k),oo);

```



```

353         Xfc=deval(sol,0:dt:dt_ana);
354         diff=Xsfc-Xfc;
355         sprintf('Verhältnis:%f\n',norm(diff(:,dt_ana/dt+1))/norm(diff
           (:,1)))
356         sprintf('%d normB %f \n',i,norm(B))
357         disp('-----')
358     end
359 end

360
361 k=k+1; %Zaehler, Index fuer Analysen
362
363 x_obs(:,k)=Xnr(:,i+1)+xo(:,k); %Beobachtung
364
365 %Kalman-Filter
366 x(:,k)=Xn(:,i+1);
367 % P=A'*P*A+Q;
368 P=(E-A)'*P*(E-A)+A'*R*A+Q;
369 trace_P_before_update(k)=trace(P);
370 S=P+R;
371 K=P/S;
372 % K(3,3)=0;
373 x(:,k)=x(:,k)+K*(x_obs(:,k)-x(:,k));
374 x(3,k)=Xn(3,1);
375 P=P-K*P;
376 P(3,:)=0;
377 P(:,3)=0;
378 trace_P_after_update(:,k)=trace(P);
379 % disp(sprintf('%d condA %f normP:%f norm(P-P):%f\n',i,cond(A),norm(P),
norm(P-P')))
380 % disp(sprintf('%d normA %f \n',i,norm(A)))
381
382 %Updateschritt
383 x_fc(:,k)=Xn(:,i+1);
384 Xn(:,i+1)=x(:,k);
385
386 %Berechnung der singulaeren Vektoren
387 [~,~,V1]=svd(norm1(Xn(:,i))*A);
388 V1(3,:)=0;
389 [~,~,V2]=svd(norm2(Xn(:,i))*A);
390 V2(3,:)=0;

```

```

391     [~,~,V3]=svd(norm3(Xn(:,i))*A);
392     V3(3,:)=0;
393
394     %Winkel zwischen den verschiedenen ersten singulären Vektoren
395     angle_12(k,1)=angle(V1(:,1),V2(:,1));
396     angle_13(k,1)=angle(V1(:,1),V3(:,1));
397     angle_23(k,1)=angle(V2(:,1),V3(:,1));
398
399     %Erstellung der Störungen
400     sigma=sqrt(trace_P_after_update(:,k));
401     Alpha=Alphastd*sigma;
402     % Alpha=random('norm',0,sigma,2,n);
403     % Alpha(Alpha>3*sigma)=3*sigma;
404     % Alpha(Alpha<-3*sigma)=-3*sigma;
405     % emp_fac=1;
406
407     for j=1:n;
408
409         v1(:,j,k)=emp_fac.*(Alpha(1,j).*V1(:,1)+Alpha(2,j).*V1(:,2)); %Lk
410         % der ersten beiden sing. Vek.
411         x_pert1(:,j,k)=v1(:,j,k)+x(:,k);
412         x_pert1(3,j,k)=Xn(3,1); %C soll nicht gestört werden
413
414         v2(:,j,k)=emp_fac.*(Alpha(1,j).*V2(:,1)+Alpha(2,j).*V2(:,2));
415         x_pert2(:,j,k)=v2(:,j,k)+x(:,k);
416         x_pert2(3,j,k)=Xn(3,1);
417
418         v3(:,j,k)=emp_fac.*(Alpha(1,j).*V3(:,1)+Alpha(2,j).*V3(:,2));
419         x_pert3(:,j,k)=v3(:,j,k)+x(:,k);
420         x_pert3(3,j,k)=Xn(3,1);
421
422         %Runge-Kutta Verfahren für Störungen:
423         for fcs=1:num_fc_steps
424
425             sol1=ode45(@forcing,[0 fc_steps(fcs)],...
426                 x_pert1(:,j,k));
427             x_pert_end1(:,j,k,fcs)=deval(sol1,fc_steps(fcs));
428             x_pert_end1(3,j,k,fcs)=Xn(3,1); %C kann sich wegen Rechengenauigkeit
429             % aendern

```

```

429         sol2=ode45(@forcing,[0 fc_steps(fcs)],...
430                     x_pert2(:,j,k));
431         x_pert_end2(:,j,k,fcs)=deval(sol2,fc_steps(fcs));
432         x_pert_end2(3,j,k,fcs)=Xn(3,1);
433
434         sol3=ode45(@forcing,[0 fc_steps(fcs)],...
435                     x_pert3(:,j,k));
436         x_pert_end3(:,j,k,fcs)=deval(sol3,fc_steps(fcs));
437         x_pert_end3(3,j,k,fcs)=Xn(3,1);
438
439     end
440 end
441 A=eye(num_dim);    %Resolvente zurücksetzen
442 end
443 end
444
445 for m=1:num_dim
446     Lyapunov_spec(m,1)=sum(Lyapunov(m,:))/i;
447 end
448
449 fprintf('Lyapunov Spektrum:\n');
450 fprintf('%f\n',Lyapunov_spec);
451
452 if stoercheck_yn==1
453     norm_v1_max=max(norm_v1);
454     disp(norm_v1_max)
455 end
456
457 if imagcheck_yn==1
458     disp('Anzahl der imaginär gewordenen Normen')
459     disp(u)
460 end
461
462 end
463
464 %%
465 %-----Verifikation-----
466 for Verifikation=1:1
467
468     %Zuschneiden der Zeitreihen

```

```

469 for fcs=1:num_fc_steps
470     to_cut_front(fcs)=(max(fc_steps)-fc_steps(fcs))/dt_ana+1;
471     to_cut_back(fcs)=t_end/dt_ana-(fc_steps(fcs)-min(fc_steps))/dt_ana-1;
472
473     x_pert_end_cut1(:, :, :, fcs)=x_pert_end1(:, :, to_cut_front(fcs):to_cut_back(
         fcs), fcs);
474     x_pert_end_cut2(:, :, :, fcs)=x_pert_end2(:, :, to_cut_front(fcs):to_cut_back(
         fcs), fcs);
475     x_pert_end_cut3(:, :, :, fcs)=x_pert_end3(:, :, to_cut_front(fcs):to_cut_back(
         fcs), fcs);
476 end
477 %Achtung x_obs_cut bedeutet die upgedatete Beobachtung (=Analyse)
478 x_obs_cut=x(:, max(to_cut_front)+1:max(to_cut_back)+1);
479 x_cut=x(:, max(to_cut_front)+1:max(to_cut_back)+1);
480 x_fc_cut=x_fc(:, max(to_cut_front)+1:max(to_cut_back)+1);
481 t_ana_cut=t_ana(:, max(to_cut_front)+1:max(to_cut_back)+1);
482 trace_P_before_update_cut=trace_P_before_update(:, max(to_cut_front)+1:max(
    to_cut_back)+1);
483 %trace_P_before_after_cut=trace_P_after_update(:, max(to_cut_front)+1:max(
    to_cut_back)+1);
484
485 %RMS
486 for m=1:num_dim
487     for kk=1:num_fc
488         for_rms(m, kk)=(x_fc_cut(m, kk)-x_cut(m, kk)).^2;
489     end
490     rms(m, 1)=sqrt(1/num_fc*sum(for_rms(m, :)));
491 end
492
493 for fcs=1:num_fc_steps
494
495 %Ensemblespread
496 for m=1:num_dim
497     for kk=1:num_fc
498         x_pert_end_mean1(m, kk, fcs)=sum(x_pert_end_cut1(m, :, kk, fcs))/n; %
            Ensemblemittel
499         x_pert_end_mean2(m, kk, fcs)=sum(x_pert_end_cut2(m, :, kk, fcs))/n;
500         x_pert_end_mean3(m, kk, fcs)=sum(x_pert_end_cut3(m, :, kk, fcs))/n;
501         for j=1:n;

```

```

502         for_spread1(m,j, kk, fcs)=(x_pert_end_cut1(m,j, kk, fcs)-
           x_pert_end_mean1(m, kk, fcs)).^2;
503         for_spread2(m,j, kk, fcs)=(x_pert_end_cut2(m,j, kk, fcs)-
           x_pert_end_mean2(m, kk, fcs)).^2;
504         for_spread3(m,j, kk, fcs)=(x_pert_end_cut3(m,j, kk, fcs)-
           x_pert_end_mean3(m, kk, fcs)).^2;
505     end
506     % Ein Freiheitsgrad weniger, weil Mittelwert schon geschätzt
507     spread1(m, kk, fcs)=(sum(for_spread1(m, :, kk, fcs))/(n-1));
508     spread2(m, kk, fcs)=(sum(for_spread2(m, :, kk, fcs))/(n-1));
509     spread3(m, kk, fcs)=(sum(for_spread3(m, :, kk, fcs))/(n-1));
510
511 end
512
513 end
514
515 for kk=1:num_fc
516     spread_norm1(kk, fcs)=norm(spread1(:, kk, fcs)); %erster Wert für t=2*dt_ana
517     % spread_norm2(kk, fcs)=norm(spread2(:, kk, fcs));
518     % spread_norm3(kk, fcs)=norm(spread3(:, kk, fcs));
519
520     for_rms_norm(kk)=norm(for_rms(:, kk)); %for_rms: erster Wert für t=dt_ana
521                                     % deswegen um einen Zeitschritt
522                                     % versetzt damit passend zum
523                                     % spread
524 end
525
526 %For Brier Score
527 for m=1:num_dim
528     for kk=1:num_fc
529         if x_obs_cut(m, kk)>0
530             x_obs_yn(m, kk)=1;
531         end
532         fc_prob1(m, kk, fcs)=sum(x_pert_end_cut1(m, :, kk, fcs)>0)/n; %pk %
           vorhergesagte Wahrscheinlichkeit
533         fc_prob2(m, kk, fcs)=sum(x_pert_end_cut2(m, :, kk, fcs)>0)/n; %pk
534         fc_prob3(m, kk, fcs)=sum(x_pert_end_cut3(m, :, kk, fcs)>0)/n; %pk
535     end
536 end
537 for_BS1(:, :, fcs)=(fc_prob1(:, :, fcs)-x_obs_yn(:, :)).^2;

```

```

538 for_BS2(:, :, fcs)=(fc_prob2(:, :, fcs)-x_obs_yn(:, :)).^2;
539 for_BS3(:, :, fcs)=(fc_prob3(:, :, fcs)-x_obs_yn(:, :)).^2;
540
541
542 %For Reliability, Resolution, Uncertainty
543 fc_prob_lb=0:0.1:0.9; %Grenzen fuer vorhergesagte Wahrscheinlichkeiten
544 fc_prob_ub=fc_prob_lb+0.1;
545 fc_prob_ub(10)=1.01;
546 fc_class_middle=0.05:0.1:0.95;
547
548 for s=1:10    %zaehlt die Vorhersageklassen durch
549     for m=1:num_dim    %Index der Komponenten A bis F
550         for kk=1:num_fc    %Index Analyseteitschritte
551
552             %Wenn vorhergesagte Wahrscheinlichkeit in best. Klasse faellt:
553             if (fc_prob1(m,kk,fcs)) >= (fc_prob_lb(1,s)) && (fc_prob1(m,kk,
                    fcs)) < (fc_prob_ub(1,s))
554                 num_pos_fc1(m,s,fcs)=num_pos_fc1(m,s,fcs)+1; %Ni
555
556             %Wenn Ereignis wirklich eintritt
557             if x_obs_cut(m,kk)>0
558                 hit1(m,s,fcs)=hit1(m,s,fcs)+1;
559             end
560         end
561
562         if (fc_prob2(m,kk,fcs)) >= (fc_prob_lb(1,s)) && (fc_prob2(m,kk,
                    fcs)) < (fc_prob_ub(1,s))
563             num_pos_fc2(m,s,fcs)=num_pos_fc2(m,s,fcs)+1; %Ni
564             if x_obs_cut(m,kk)>0
565                 hit2(m,s,fcs)=hit2(m,s,fcs)+1;
566             end
567         end
568
569         if (fc_prob3(m,kk,fcs)) >= (fc_prob_lb(1,s)) && (fc_prob3(m,kk,
                    fcs)) < (fc_prob_ub(1,s))
570             num_pos_fc3(m,s,fcs)=num_pos_fc3(m,s,fcs)+1; %Ni
571             if x_obs_cut(m,kk)>0
572                 hit3(m,s,fcs)=hit3(m,s,fcs)+1;
573             end
574         end

```

```

575     end
576
577     num_pos_fc_for_cond1(m,s,fcs)=num_pos_fc1(m,s,fcs);
578     if (num_pos_fc_for_cond1(m,s,fcs) == 0) %fuer cond_prob,da 0/0=NaN ->
579         0/1=0 erzwungen
580         num_pos_fc_for_cond1(m,s,fcs)=1;    %wenn num_pos_fc=0 automatisch
581         hit=0
582     end
583     cond_prob1(m,s,fcs)=hit1(m,s)/num_pos_fc_for_cond1(m,s); %oi_quer %
584     bedingte Wahrscheinlichkeit
585     clim_prob(m,1)=sum(x_obs_yn(m,:))/num_fc; %o_quer %klimatologische
586     Wahrscheinlichkeit
587     for_BS_rel1(m,s,fcs)=num_pos_fc1(m,s,fcs)*(fc_class_middle(s)-
588         cond_prob1(m,s,fcs)).^2;
589     for_BS_res1(m,s,fcs)=num_pos_fc1(m,s,fcs)*(cond_prob1(m,s,fcs)-
590         clim_prob(m,1)).^2;
591
592     num_pos_fc_for_cond2(m,s,fcs)=num_pos_fc2(m,s,fcs);
593     if (num_pos_fc_for_cond2(m,s,fcs) == 0)
594         num_pos_fc_for_cond2(m,s,fcs)=1;
595     end
596     cond_prob2(m,s,fcs)=hit2(m,s,fcs)/num_pos_fc_for_cond2(m,s,fcs); %
597     oi_quer
598     for_BS_rel2(m,s,fcs)=num_pos_fc2(m,s,fcs)*(fc_class_middle(s)-
599         cond_prob2(m,s,fcs)).^2;
600     for_BS_res2(m,s,fcs)=num_pos_fc2(m,s,fcs)*(cond_prob2(m,s,fcs)-
601         clim_prob(m,1)).^2;
602
603     num_pos_fc_for_cond3(m,s,fcs)=num_pos_fc3(m,s,fcs);
604     if (num_pos_fc_for_cond3(m,s,fcs) == 0)
605         num_pos_fc_for_cond3(m,s,fcs)=1;
606     end
607     cond_prob3(m,s,fcs)=hit3(m,s,fcs)/num_pos_fc_for_cond3(m,s,fcs); %
608     oi_quer
609     for_BS_rel3(m,s,fcs)=num_pos_fc3(m,s,fcs)*(fc_class_middle(s)-
610         cond_prob3(m,s,fcs)).^2;
611     for_BS_res3(m,s,fcs)=num_pos_fc3(m,s,fcs)*(cond_prob3(m,s,fcs)-
612         clim_prob(m,1)).^2;
613
614 end
end

```

```

603
604 %Brier Score, Reliability, Resolution, Uncertainty
605 for m=1:num_dim
606     BS1(m,fcs)=1/(num_fc)*sum(for_BS1(m,:,fcs)); %BS
607     BS_rel1(m,fcs)=1/(num_fc)*sum(for_BS_rel1(m,:,fcs)); %BS Reliability
608     BS_res1(m,fcs)=1/(num_fc)*sum(for_BS_res1(m,:,fcs)); %BS Resolution
609     BS_unc1(m,fcs)=clim_prob(m,1)*(1-clim_prob(m,1)); %BS Uncertainty
610
611     BS2(m,fcs)=1/(num_fc)*sum(for_BS2(m,:,fcs));
612     BS_rel2(m,fcs)=1/(num_fc)*sum(for_BS_rel2(m,:,fcs));
613     BS_res2(m,fcs)=1/(num_fc)*sum(for_BS_res2(m,:,fcs));
614
615     BS3(m,fcs)=1/(num_fc)*sum(for_BS3(m,:,fcs));
616     BS_rel3(m,fcs)=1/(num_fc)*sum(for_BS_rel3(m,:,fcs));
617     BS_res3(m,fcs)=1/(num_fc)*sum(for_BS_res3(m,:,fcs));
618 end
619 BS_unc2=BS_unc1;
620 BS_unc3=BS_unc1;
621
622 %ROC-Diagramm
623 for m=1:num_dim
624     for h=1:size(th,2)
625         for kk=1:num_fc
626
627             %norm1
628             if fc_prob1(m,kk,fcs)>th(h) && x_obs_yn(m,kk)>0
629                 Hit1(m,h,fcs)=Hit1(m,h,fcs)+1;
630             end
631
632             if fc_prob1(m,kk,fcs)>th(h) && x_obs_yn(m,kk)<1
633                 False1(m,h,fcs)=False1(m,h,fcs)+1;
634             end
635
636             if fc_prob1(m,kk,fcs)<th(h) && x_obs_yn(m,kk)>0
637                 Miss1(m,h,fcs)=Miss1(m,h,fcs)+1;
638             end
639
640             if fc_prob1(m,kk,fcs)<th(h) && x_obs_yn(m,kk)<1
641                 Zero1(m,h,fcs)=Zero1(m,h,fcs)+1;
642             end

```



```

643
644 %norm2
645 if fc_prob2(m, kk, fcs) > th(h) && x_obs_yn(m, kk) > 0
646     Hit2(m, h, fcs) = Hit2(m, h, fcs) + 1;
647 end
648
649 if fc_prob2(m, kk, fcs) > th(h) && x_obs_yn(m, kk) < 1
650     False2(m, h, fcs) = False2(m, h, fcs) + 1;
651 end
652
653 if fc_prob2(m, kk, fcs) < th(h) && x_obs_yn(m, kk) > 0
654     Miss2(m, h, fcs) = Miss2(m, h, fcs) + 1;
655 end
656
657 if fc_prob2(m, kk, fcs) < th(h) && x_obs_yn(m, kk) < 1
658     Zero2(m, h, fcs) = Zero2(m, h, fcs) + 1;
659 end
660
661 %norm3
662 if fc_prob3(m, kk, fcs) > th(h) && x_obs_yn(m, kk) > 0
663     Hit3(m, h, fcs) = Hit3(m, h, fcs) + 1;
664 end
665
666 if fc_prob3(m, kk, fcs) > th(h) && x_obs_yn(m, kk) < 1
667     False3(m, h, fcs) = False3(m, h, fcs) + 1;
668 end
669
670 if fc_prob3(m, kk, fcs) < th(h) && x_obs_yn(m, kk) > 0
671     Miss3(m, h, fcs) = Miss3(m, h, fcs) + 1;
672 end
673
674 if fc_prob3(m, kk, fcs) < th(h) && x_obs_yn(m, kk) < 1
675     Zero3(m, h, fcs) = Zero3(m, h, fcs) + 1;
676 end
677 end
678
679 POD1(m, h, fcs) = Hit1(m, h, fcs) / (Hit1(m, h, fcs) + Miss1(m, h, fcs)); %Probability
    of detection
680
    POFD1(m, h, fcs) = False1(m, h, fcs) / (False1(m, h, fcs) + Zero1(m, h, fcs)); %
    Probability of false detection

```

```

681
682     POD2(m,h,fcs)=Hit2(m,h,fcs)/(Hit2(m,h,fcs)+Miss2(m,h,fcs));
683     POFD2(m,h,fcs)=False2(m,h,fcs)/(False2(m,h,fcs)+Zero2(m,h,fcs));
684
685     POD3(m,h,fcs)=Hit3(m,h,fcs)/(Hit3(m,h,fcs)+Miss3(m,h,fcs));
686     POFD3(m,h,fcs)=False3(m,h,fcs)/(False3(m,h,fcs)+Zero3(m,h,fcs));
687
688     end
689 end
690
691 %Speichern der BS-Werte in txt-file
692 BS_Mat(:, :, fcs)=[BS1(:, fcs), BS2(:, fcs), BS3(:, fcs)]';
693 BS_rel_Mat(:, :, fcs)=[BS_rel1(:, fcs), BS_rel2(:, fcs), BS_rel3(:, fcs)]';
694 BS_res_Mat(:, :, fcs)=[BS_res1(:, fcs), BS_res2(:, fcs), BS_res3(:, fcs)]';
695 BS_unc_Mat(:, :, fcs)=[BS_unc1(:, fcs), BS_unc2(:, fcs), BS_unc3(:, fcs)]';
696
697 %Vergleich der Normen
698 Diff12(:, 1)=BS1(:, fcs)-BS2(:, fcs);
699 Diff13(:, 1)=BS1(:, fcs)-BS3(:, fcs);
700 norm2_better_norm1=0;
701 norm3_better_norm1=0;
702
703 for m=1:num_dim
704     if Diff12(m,1)>0
705         Diff12_01(m,1)=1;
706         norm2_better_norm1=norm2_better_norm1+1;
707     end
708
709     if Diff13(m,1)>0
710         Diff13_01(m,1)=1;
711         norm3_better_norm1=norm3_better_norm1+1;
712     end
713
714     if Diff12(m,1)<0
715         Diff12_01(m,1)=-1;
716         norm2_better_norm1=norm2_better_norm1+1;
717     end
718
719     if Diff13(m,1)<0
720         Diff13_01(m,1)=-1;

```

```

721         norm3_better_norm1=norm3_better_norm1+1;
722     end
723 end
724 Diff11_01=zeros(num_dim,1);
725 Diff=[Diff11_01(:,1),Diff12_01(:,1), Diff13_01(:,1)]';
726
727 end
728
729 %Überprüfung von BS:
730 sollte_null_sein=BS1-BS_rel1+BS_res1-BS_unc1;
731 disp('ich sollte Null sein:')
732 disp(sollte_null_sein)
733
734 fid=fopen('out.txt','w');
735
736 fprintf(fid,'-----Parameterwahl----- \n');
737 fprintf(fid,'dt= %0.3f \nt_end= %d \ndt_ana= %0.3f \nn=%d \n',dt,t_end,dt_ana,n
    );
738 fprintf(fid,'c1= %0.3f \nc2= %0.3f \nemp_fac= %0.3f \n',c1,c2,emp_fac);
739
740 fprintf(fid,'\n-----Verifikation----- \n\n');
741
742 fprintf(fid,'Norm 1 = H-Norm\n');
743 fprintf(fid,'Norm 2 = H-Norm + c1 * C-Norm\n');
744 fprintf(fid,'Norm 2 = H-Norm + c1 * C-Norm + c2 * S-Norm\n\n\n');
745
746 fprintf(fid,' \t Norm 1 \t\t Norm 2 \t\t Norm 3');
747
748 fprintf(fid,'\n\nDiff zu norm1: 1 bedeutet besser,0 gleich und -1 schlechter
    als norm1\n');
749
750 for fcs=1:num_fc_steps
751     fprintf(fid,'\nfc_step= %0.3f\n\n',fc_steps(fcs));
752     fprintf(fid,'%14.2f %14.2f %14.2f\n',Diff(:,:));
753 end
754
755 fprintf(fid,'\nBS\n');
756 for fcs=1:num_fc_steps
757     fprintf(fid,'\nfc_step= %0.3f\n\n',fc_steps(fcs));
758     fprintf(fid,'%14.9f %14.9f %14.9f\n',BS_Mat(:,1:num_dim,fcs));

```

```

759 end
760
761 fprintf(fid, '\n\nBS_rel\n');
762 for fcs=1:num_fc_steps
763     fprintf(fid, '\nfc_step= %0.3f\n\n', fc_steps(fcs));
764     fprintf(fid, '%14.9f %14.9f %14.9f\n', BS_rel_Mat(:, 1:num_dim, fcs));
765 end
766
767 fprintf(fid, '\n\nBS_res\n');
768 for fcs=1:num_fc_steps
769     fprintf(fid, '\nfc_step= %0.3f\n\n', fc_steps(fcs));
770     fprintf(fid, '%14.9f %14.9f %14.9f\n', BS_res_Mat(:, 1:num_dim, fcs));
771 end
772
773 fprintf(fid, '\n\nBS_unc\n');
774 for fcs=1:num_fc_steps
775     fprintf(fid, '\nfc_step= %0.3f\n\n', fc_steps(fcs));
776     fprintf(fid, '%14.5f %14.5f %14.5f\n', BS_unc_Mat(:, 1:num_dim, fcs));
777 end
778
779 fclose(fid);
780
781 end
782
783 save('rbc_solution')
784 disp('u=') %anzahl der imaginären Gewichtungsmatrizen
785 disp(u)
786
787 %%
788 %-----Plots-----
789 for Plots=1:1
790
791     for_axes=['A' 'B' 'C' 'D' 'E' 'F'];
792
793     for fcs=1:num_fc_steps
794
795         if plot_yn==1
796             delete out.ps
797
798             %Phasenraumplots 2d

```

```

799 fig1=figure('Units','normalized','Position',[0 0 1 1]);
800 set(fig1,'name','Phasenraumdiagramme 2d','numbertitle','off')
801
802 l=1;
803 for xx=1:6
804     for yy=1:xx-1
805         subplot(4,3,l)
806         if xx~=3 && yy~=3
807             plot(Xnr(xx,:),Xnr(yy,:))
808             xlabel(for_axes(xx))
809             ylabel(for_axes(yy))
810             title(sprintf('Phasenraum:\n Komponenten %s und %s',for_axes(yy),
                        for_axes(xx)))
811             l=l+1;
812         end
813     end
814 end
815 print(gcf,'-dpsc','-append','out.ps')
816 close(gcf)
817
818
819 %Phasenraumplots 3d
820 fig2=figure('Units','normalized','Position',[0 0 1 1]);
821 set(fig2,'name','Phasenraumdiagramme 3d','numbertitle','off')
822 l=1;
823 for xx=1:6
824     for yy=1:xx-1
825         for zz=1:yy-1
826             subplot(4,3,l)
827             if xx~=3 && yy~=3 && zz~=3
828                 plot3(Xnr(xx,:),Xnr(yy,:),Xnr(zz,:))
829                 grid on
830                 xlabel(for_axes(xx))
831                 ylabel(for_axes(yy))
832                 zlabel(for_axes(zz))
833                 title(sprintf('Phasenraum:\n Komponenten %s, %s und %s',for_axes(zz),
                        for_axes(yy),for_axes(xx)))
834                 l=l+1;
835             end
836         end
837     end
838 end

```

```

837     end
838 end
839 print(gcf, '-dpsc', '-append', 'out.ps')
840 %close(gcf)
841
842
843 %Temperatur- und Vorticityfelder
844 for kk=linspace(1,num_fc,2)
845     %kk=1:t_end/dt_ana-1;
846
847     Phi11=b*x_cut(1,kk);
848     phi11=b*x_cut(2,kk);
849     phi02=c*x_cut(3,kk);
850     theta11=e*x_cut(4,kk);
851     Theta11=e*x_cut(5,kk);
852     Theta02=f*x_cut(6,kk);
853
854     x_feld=0:0.01:2;
855     z_feld=0:0.01:2;
856     [xx,zz]=meshgrid(x_feld,z_feld);
857
858     Phi=(Phi11*sin(a*pi*xx+pi)+phi11*cos(a*pi*xx+pi)).*sin(pi*zz)...
859         +phi02*cos(0*xx).*sin(2*pi*zz);
860     Theta=(theta11*sin(a*pi*xx+pi)+Theta11*cos(a*pi*xx+pi)).*sin(pi*zz)...
861         +Theta02*cos(0).*sin(2*pi*zz);
862
863     fig3=figure('Units','normalized','Position',[0 0.3 1 0.45]);
864     set(fig3,'name','Stroemungs- und Temperaturfelder','numbertitle','off')
865     subplot(1,2,1)
866     contour(xx,zz,Phi)
867     t_for_title=(kk+max(fc_steps)/dt_ana)*dt_ana;
868     title(sprintf('Phi bei t=%g',t_for_title))
869     xlabel('x')
870     ylabel('z')
871     colorbar
872     subplot(1,2,2)
873     contour(xx,zz,Theta)
874     title(sprintf('Theta bei t=%g',t_for_title))
875     xlabel('x')
876     ylabel('z')

```

```

877     colorbar
878     print(gcf,'-dpasc','-append','out.ps')
879     %close(gcf)
880 end
881
882
883 %Entwickelte Komponenten mit Beobachtungen, Störungen, Ensembles
884 fig4=figure('Units','normalized','Position',[0 0 1 1]);
885 set(fig4,'name','Entwicklung aller Komponenten - H-norm','numbertitle','off')
886 for m=1:num_dim
887     subplot(3,2,m)
888     hold on
889     plot(t,Xn(m,:));
890     plot(t_ana,x_obs(m,:), 'ro'); %Beobachtung
891     plot(t_ana,x(m,:), 'co'); %nach Update
892     plot(t_ana,x_fc(m,:), 'o') %Vorhersage vor Update
893     for kk=1:k;
894         plot(ones(1,n).*dt_ana*kk,x_pert1(m,:,kk), 'k+') %Stoerungen
895         plot(ones(1,n).*dt_ana*kk+fc_steps(fcs),x_pert_end1(m,:,kk), 'gx') %
            integrierte Stoerungen
896         for j=1:n;
897             line([dt_ana*kk dt_ana*kk+fc_steps(fcs)], [x_pert1(m,j,kk)
            x_pert_end1(m,j,kk)], 'Color', 'g')
898         end
899     end
900     xlim([0 t_end])
901     xlabel('t')
902     ylabel(for_axes(m))
903     title(sprintf('Entwicklung von %s',for_axes(m)))
904 end
905 print(gcf,'-dpasc','-append','out.ps')
906 %close(gcf)
907
908
909 %norm1: Entwickelte A-Komponente mit Beobachtungen, Störungen, Ensembles
910 fig5_1=figure('Units','normalized','Position',[0 0 1 1]);
911 set(fig5_1,'name','Entwicklung von Komponente A fuer H-Norm','numbertitle','off
    ')
912 hold on
913 m=1;

```

```

914 h1=plot(t,Xnr(m,:), 'm-');
915 h2=plot(t,Xn(m,:));
916 for kk=1:k;
917     h3=plot(ones(1,n).*dt_ana*kk,x_pert1(m,:,kk), 'k+');
918     h4=plot(ones(1,n).*dt_ana*kk+fc_steps(fcs),x_pert_end1(m,:,kk), 'g+');
919     for j=1:n;
920         line([dt_ana*kk dt_ana*kk+fc_steps(fcs)], [x_pert1(m,j,kk) x_pert_end1(m,
921             j,kk)], 'Color', 'g')
922     end
923 end
924 h5=plot(t_ana,x(m,:), 'co');
925 h6=plot(t_ana,x_fc(m,:), 'o');
926 h7=plot(t_ana,x_obs(m,:), 'ro');
927 legend([h1,h2,h3,h4,h5,h6,h7],...
928     'freier fc', 'fc mit updates',...
929     'Störungen', 'integr. Störungen',...
930     'Analyse', 'nichtlin fc', 'Beob.')
931 xlim([0 t_end])
932 xlabel('t')
933 ylabel('A')
934 title('Entwicklung von Komponente A fuer norm1')
935 print(gcf, '-dpsc', '-append', 'out.ps')
936 %close(gcf)
937 saveas(gcf, 'ergebnisse/KompA_Zeitreihe.fig')
938
939 %norm2: Entwickelte A-Komponente mit Beobachtungen, Störungen, Ensembles
940 fig5_1=figure('Units', 'normalized', 'Position', [0 0 1 1]);
941 set(fig5_1, 'name', 'Entwicklung von Komponente A fuer norm2', 'numbertitle', 'off'
942     )
943 hold on
944 m=1;
945 h1=plot(t,Xnr(m,:), 'm-');
946 h2=plot(t,Xn(m,:));
947 for kk=1:k;
948     h3=plot(ones(1,n).*dt_ana*kk,x_pert2(m,:,kk), 'k+');
949     h4=plot(ones(1,n).*dt_ana*kk+fc_steps(fcs),x_pert_end2(m,:,kk), 'g+');
950     for j=1:n;
951         line([dt_ana*kk dt_ana*kk+fc_steps(fcs)], [x_pert2(m,j,kk) x_pert_end2(m,
952             j,kk)], 'Color', 'g')

```



```

951     end
952 end
953 h5=plot(t_ana,x(m,:), 'co');
954 h6=plot(t_ana,x_fc(m,:), 'o');
955 h7=plot(t_ana,x_obs(m,:), 'ro');
956 legend([h1,h2,h3,h4,h5,h6,h7],...
957     'freier fc', 'fc mit updates',...
958     'Störungen', 'integr. Störungen',...
959     'Analyse', 'nichtlin fc', 'Beob.')
```

960 xlim([0 t\_end])

961 xlabel('t')

962 ylabel('A')

963 title('Entwicklung von Komponente A fuer norm2')

964 print(gcf, '-dpsc', '-append', 'out.ps')

965 %close(gcf)

966

967

968 %norm3: Entwickelte A-Komponente mit Beobachtungen, Störungen, Ensembles

969 fig5\_1=figure('Units', 'normalized', 'Position', [0 0 1 1]);

970 set(fig5\_1, 'name', 'Entwicklung von Komponente A fuer norm3', 'numbertitle', 'off'

971 )

972 hold on

973 m=1;

974 h1=plot(t, Xnr(m,:), 'm-');

975 h2=plot(t, Xn(m,:));

976 for kk=1:k;

977 h3=plot(ones(1,n).\*dt\_ana\*kk,x\_pert3(m,:,kk), 'k+');

978 h4=plot(ones(1,n).\*dt\_ana\*kk+fc\_steps(fcs),x\_pert\_end3(m,:,kk), 'g+');

979 for j=1:n;

980 line([dt\_ana\*kk dt\_ana\*kk+fc\_steps(fcs)], [x\_pert3(m,j,kk) x\_pert\_end3(m,

981 j,kk)], 'Color', 'g')

982 end

983 end

984 h5=plot(t\_ana,x(m,:), 'co');

985 h6=plot(t\_ana,x\_fc(m,:), 'o');

986 h7=plot(t\_ana,x\_obs(m,:), 'ro');

987 legend([h1,h2,h3,h4,h5,h6,h7],...
988 'freier fc', 'fc mit updates',...
989 'Störungen', 'integr. Störungen',...
990 'Analyse', 'nichtlin fc', 'Beob.')

```

989 xlim([0 t_end])
990 xlabel('t')
991 ylabel('A')
992 title('Entwicklung von Komponente A fuer norm2')
993 print(gcf,'-dpsc','-append','out.ps')
994 %close(gcf)
995
996
997 % Zeitliche Entwicklung der Normen, absolut
998 fig5_A=figure('Units','normalized','Position',[0 0 1 1]);
999 set(fig5_A,'name','Zeitliche Entwicklung der Normen, absolut','numbertitle','
    off')
1000 subplot(2,2,1)
1001 hold on
1002 plot(t,E_norm_Xnr)
1003 title('Zeitliche Entwicklung Euler-Norm')
1004 xlim([0 t_end])
1005 xlabel('t')
1006 ylabel('berechnete Norm')
1007 subplot(2,2,2)
1008 hold on
1009 plot(t,H_norm_Xnr)
1010 title('Zeitliche Entwicklung H-Norm')
1011 xlim([0 t_end])
1012 xlabel('t')
1013 ylabel('berechnete Norm')
1014 subplot(2,2,3)
1015 hold on
1016 plot(t,C_norm_Xnr)
1017 title('Zeitliche Entwicklung C-Norm')
1018 xlim([0 t_end])
1019 xlabel('t')
1020 ylabel('berechnete Norm')
1021 subplot(2,2,4)
1022 hold on
1023 plot(t,S_norm_Xnr)
1024 title('Zeitliche Entwicklung S-Norm')
1025 xlim([0 t_end])
1026 xlabel('t')
1027 ylabel('berechnete Norm')

```

```

1028 print(gcf, '-dpsc', '-append', 'out.ps')
1029 %close(gcf)
1030
1031
1032 %Zeitliche Entwicklung der Normen, normiert
1033 fig5_B=figure('Units','normalized','Position',[0 0 1 1]);
1034 set(fig5_B,'name','Zeitliche Entwicklung der Normen, normiert auf Startwerte','
    numbertitle','off')
1035 subplot(2,2,1)
1036 hold on
1037 line([0 t_end],[1 1],'Color','r','Linewidth',2.5)
1038 plot(t,E_norm_Xnr./E_norm_Xnr(1,1),'Linewidth',1)
1039 title('Zeitliche Entwicklung E-Norm, normiert auf Startwert')
1040 xlim([0 t_end])
1041 ylim([0 4])
1042 subplot(2,2,2)
1043 hold on
1044 line([0 t_end],[1 1],'Color','r','Linewidth',2.5)
1045 plot(t,H_norm_Xnr./H_norm_Xnr(1,1),'Linewidth',1)
1046 title('Zeitliche Entwicklung norm1, normiert auf Startwert')
1047 xlim([0 t_end])
1048 ylim([0 4])
1049 subplot(2,2,3)
1050 hold on
1051 line([0 t_end],[1 1],'Color','r','Linewidth',2.5)
1052 plot(t,C_norm_Xnr./C_norm_Xnr(1,1),'Linewidth',1)
1053 title('Zeitliche Entwicklung norm2, normiert auf Startwert')
1054 xlim([0 t_end])
1055 ylim([0 4])
1056 subplot(2,2,4)
1057 hold on
1058 line([0 t_end],[1 1],'Color','r','Linewidth',2.5)
1059 plot(t,S_norm_Xnr./S_norm_Xnr(1,1),'Linewidth',1)
1060 title('Zeitliche Entwicklung norm3, normiert auf Startwert')
1061 xlim([0 t_end])
1062 ylim([0 4])
1063 print(gcf, '-dpsc', '-append', 'out.ps')
1064 %close(gcf)
1065
1066

```

```

1067 %Spur von P
1068 fig6=figure('Units','normalized','Position',[0 0 1 1]);
1069 set(fig6,'name','Spur von P','numbertitle','off')
1070 hold on
1071 plot(t_ana(1,1:t_end/dt_ana),trace_P_before_update(1,1:t_end/dt_ana))
1072 title('Spur von P')
1073 xlabel('t')
1074 ylabel('trace(P)')
1075 print(gcf,'-dpsc','-append','out.ps')
1076 %close(gcf)
1077
1078 %Überprüfe generierten Beobachtungen, da wenn dt_ana zu groß zu starke
1079 %Abweichungen auftreten
1080 fig7_A=figure('Units','normalized','Position',[0 0 1 1]);
1081 set(fig7_A,'name','Generierte Störungen','numbertitle','off')
1082
1083 %Aus freier Vorhersage generierte Beobachtungen
1084 subplot(3,1,1)
1085 comp=1;
1086 hold on
1087 plot(t,Xnr(comp,:), 'm-');
1088 plot(t_ana,x_obs(comp,:), 'ro');
1089 xlabel('t')
1090 ylabel('Komponente A')
1091 title('Freie Vorhersage und daraus generierte Beobachtungen')
1092
1093 %Unterschied zwischen freier Vorhersage und Vorhersage mit Update
1094 subplot(3,1,2)
1095 hold on
1096 plot(t,Xnr(comp,:), 'm-');
1097 plot(t,Xn(comp,:), 'b-');
1098 legend('Xnr','Xn')
1099 xlabel('t')
1100 ylabel('Komponente A')
1101 title('Freie Vorhersage und Vorhersage mit Update')
1102
1103 %Differenz zwischen freier Vorhersage und Vorhersage mit Update
1104 subplot(3,1,3)
1105 hold on
1106 plot(t,Xnr(comp,:)-Xn(comp,:), 'm-')

```

```

1107 xlabel('t')
1108 ylabel('Komponente A')
1109 title('Differenz zwischen freier Vorhersage und Vorhersage mit Update')
1110 print(gcf,'-dpasc','-append','out.ps')
1111 %close(gcf)
1112
1113
1114 %Vorhersage - Beobachtung
1115 fig7=figure('Units','normalized','Position',[0 0 1 1]);
1116 set(fig7,'name','Vorhersage - Beobachtung','numbertitle','off')
1117 for m=1:num_dim
1118     subplot(3,2,m)
1119     plot(t_ana,x_fc(m,:)-x_obs(m,:))
1120     xlabel('t')
1121     ylabel(for_axes(m))
1122     title(sprintf('Abstand Vorhersage - Beobachtung für %s',for_axes(m)))
1123 end
1124 print(gcf,'-dpasc','-append','out.ps')
1125 %%close(gcf)
1126
1127
1128 %Analyse - Beobachtung und Vorgersage - Analyse
1129 fig8=figure('Units','normalized','Position',[0 0 1 1]);
1130 set(fig8,'name','Analyse - Beobachtung und Vorgersage - Analyse','numbertitle',
    'off')
1131 for m=1:num_dim
1132     subplot(3,2,m)
1133     hold on
1134     plot(t_ana,abs(x(m,:)-x_obs(m,:)),'k')
1135     plot(t_ana,abs(x_fc(m,:)-x(m,:)),'g')
1136     legend('Abstand zur obs','Abstand zur fc')
1137     xlabel('t')
1138     ylabel(for_axes(m))
1139     title(sprintf('Abstand von der Analyse für %s',for_axes(m)))
1140 end
1141 print(gcf,'-dpasc','-append','out.ps')
1142 %%close(gcf)
1143
1144
1145 %Winkel zwischen den singulären Vektoren

```

```

1146 fig9=figure('Units','normalized','Position',[0 0 1 1]);
1147 set(fig9,'name','Winkel zwischen den singulären Vektoren','numbertitle','off')
1148
1149 subplot(3,1,1)
1150 plot(t_ana,angle_12)
1151 title('Winkel zw sing. Vektoren norm1 und norm2')
1152 xlabel('t')
1153 ylabel('Winkel')
1154
1155 subplot(3,1,2)
1156 plot(t_ana,angle_13)
1157 title('Winkel zw sing. Vektoren norm1 und norm3')
1158 xlabel('t')
1159 ylabel('Winkel')
1160
1161 subplot(3,1,3)
1162 plot(t_ana,angle_23)
1163 title('Winkel zw sing. Vektoren norm2 und norm3')
1164 xlabel('t')
1165 ylabel('Winkel')
1166
1167 print(gcf,'-dpsc','-append','out.ps')
1168 %%close(gcf)
1169
1170 end
1171 %Vergleich von rms, spread und Spur(P)
1172 fig10=figure('Units','normalized','Position',[0 0 1 1]);
1173 set(fig10,'name','rms vs spread','numbertitle','off')
1174 subplot(2,1,1)
1175 title('Vergleich von rms, spread und trace(P)')
1176 hold on
1177 plot(t_ana_cut,for_rms_norm)
1178 plot(t_ana_cut,spread_norm1(:,fcs),'g')
1179 plot(t_ana_cut,trace_P_before_update_cut,'r')
1180 legend('rms','spread','trace(P)')
1181 xlabel('t')
1182 ylabel('rms, spread bzw. trace(P)')
1183
1184 subplot(2,1,2)
1185 title('Scatter rms gegen spread')

```

```

1186 hold on
1187 plot(spread_norm1(:,fcs),for_rms_norm,'o')
1188 xlabel('spread')
1189 ylabel('rms')
1190
1191 print(gcf,'-dpasc','-append','out.ps')
1192 %close(gcf)
1193 if plot_yn==1
1194
1195 %RMS und Ensemblespread - norm1
1196 fig11_1=figure('Units','normalized','Position',[0 0 1 1]);
1197 set(fig11_1,'name','RMS und Ensemblespread - norm1','numbertitle','off')
1198 for m=1:num_dim
1199     subplot(3,2,m)
1200     hold on
1201     plot(t_ana_cut,for_rms(m,:))
1202     plot(t_ana_cut,spread1(m,:,fcs),'r')
1203     legend('rms','Ensemblespread')
1204     title(sprintf('norm1: RMS und Ensemblespread für %s',for_axes(m)))
1205     xlabel('t')
1206     ylabel('rms bzw. spread')
1207 end
1208 print(gcf,'-dpasc','-append','out.ps')
1209 close(gcf)
1210
1211 %RMS und Ensemblespread - norm2
1212 fig11_1=figure('Units','normalized','Position',[0 0 1 1]);
1213 set(fig11_1,'name','RMS und Ensemblespread - norm2','numbertitle','off')
1214 for m=1:num_dim
1215     subplot(3,2,m)
1216     hold on
1217     plot(t_ana_cut,for_rms(m,:))
1218     plot(t_ana_cut,spread2(m,:,fcs),'r')
1219     legend('rms','Ensemblespread')
1220     title(sprintf('norm2: RMS und Ensemblespread für %s',for_axes(m)))
1221     xlabel('t')
1222     ylabel('rms bzw. spread')
1223 end
1224 print(gcf,'-dpasc','-append','out.ps')
1225 close(gcf)

```

```

1226
1227
1228 %RMS und Ensemblespread - norm3
1229 fig11_1=figure('Units','normalized','Position',[0 0 1 1]);
1230 set(fig11_1,'name','RMS und Ensemblespread - norm3','numbertitle','off')
1231 for m=1:num_dim
1232     subplot(3,2,m)
1233     hold on
1234     plot(t_ana_cut,for_rms(m,:))
1235     plot(t_ana_cut,spread3(m,:,fcs),'r')
1236     legend('rms','Ensemblespread')
1237     title(sprintf('norm3: RMS und Ensemblespread für %s',for_axes(m)))
1238     xlabel('t')
1239     ylabel('rms bzw. spread')
1240 end
1241 print(gcf,'-dpsc','-append','out.ps')
1242 close(gcf)
1243
1244
1245 %Reliability_Diagramm norm1
1246 fig12_1=figure('Units','normalized','Position',[0 0 1 1]);
1247 set(fig12_1,'name','Reliability_Diagramm - norm1','numbertitle','off')
1248 for m=1:num_dim
1249     subplot(6,2,2*m-1)
1250     hold on
1251     plot(fc_class_middle,cond_prob1(m,1:10,fcs),'b*')
1252     line([0 1],[0 1],'Color','k')
1253     title(sprintf('norm1: rel für %s',for_axes(m)))
1254     xlabel('fc prob')
1255     ylabel('obs prob')
1256     subplot(6,2,2*m)
1257     hist(fc_prob1(m,:,fcs))
1258     title(sprintf('norm1: fc prob für %s',for_axes(m)))
1259     xlabel('rel fc prob')
1260 end
1261 print(gcf,'-dpsc','-append','out.ps')
1262 close(gcf)
1263
1264
1265 %Reliability_Diagramm norm2

```



```

1266 fig12_1=figure('Units','normalized','Position',[0 0 1 1]);
1267 set(fig12_1,'name','Reliability_Diagramm - norm2','numbertitle','off')
1268 for m=1:num_dim
1269     subplot(6,2,2*m-1)
1270     hold on
1271     plot(fc_class_middle,cond_prob2(m,1:10,fcs),'r*')
1272     line([0 1],[0 1],'Color','k')
1273     title(sprintf('norm2: rel für %s',for_axes(m)))
1274     xlabel('fc prob')
1275     ylabel('obs prob')
1276     subplot(6,2,2*m)
1277     hist(fc_prob2(m,:),fcs)
1278     title(sprintf('norm2: fc prob für %s',for_axes(m)))
1279     xlabel('rel fc prob')
1280 end
1281 print(gcf,'-dpsc','-append','out.ps')
1282 close(gcf)
1283
1284 %Reliability_Diagramm norm3
1285 fig12_3=figure('Units','normalized','Position',[0 0 1 1]);
1286 set(fig12_3,'name','Reliability_Diagramm - norm3','numbertitle','off')
1287 for m=1:num_dim
1288     subplot(6,2,2*m-1)
1289     hold on
1290     plot(fc_class_middle,cond_prob3(m,1:10,fcs),'b*')
1291     line([0 1],[0 1],'Color','k')
1292     title(sprintf('norm1: rel für %s',for_axes(m)))
1293     xlabel('fc prob')
1294     ylabel('obs prob')
1295     subplot(6,2,2*m)
1296     hist(fc_prob3(m,:),fcs)
1297     title(sprintf('norm1: fc prob für %s',for_axes(m)))
1298     xlabel('rel fc prob')
1299 end
1300 print(gcf,'-dpsc','-append','out.ps')
1301 close(gcf)
1302
1303
1304 %Reliability_Diagramm norm1-norm2
1305 fig12_4=figure('Units','normalized','Position',[0 0 1 1]);

```

```

1306 set(fig12_4,'name','Reliability_Diagramm - norm1-norm2','numbertitle','off')
1307 for m=1:num_dim
1308     subplot(3,2,m)
1309     hold on
1310     plot(fc_class_middle,cond_prob1(m,1:10,fcs),'b*')
1311     plot(fc_class_middle,cond_prob2(m,1:10,fcs),'r*')
1312     line([0 1],[0 1],'Color','k')
1313     title(sprintf('normH, normHC: rel für %s',for_axes(m)))
1314     xlabel('fc prob')
1315     ylabel('obs prob')
1316     legend('normH','normHC',2)
1317 end
1318 print(gcf,'-dpasc','-append','out.ps')
1319 print(gcf, '-depasc', 'ergebnisse/BS_rel vgl normH-normHC.eps');
1320 saveas(gcf,'ergebnisse/BS_rel vgl normH-normHC.fig')
1321 close(gcf)
1322
1323 %Reliability_Diagramm norm1-norm2 hist
1324 fig12_5=figure('Units','normalized','Position',[0 0 1 1]);
1325 set(fig12_5,'name','Reliability_Diagramm - norm1-norm2','numbertitle','off')
1326 for m=1:num_dim
1327     subplot(6,2,2*m-1)
1328     hold on
1329     hist(fc_prob1(m,:),fcs)
1330     title(sprintf('normH: fc prob für %s',for_axes(m)))
1331     xlabel('rel fc prob')
1332     subplot(6,2,2*m)
1333     hist(fc_prob2(m,:),fcs)
1334     title(sprintf('normHC: fc prob für %s',for_axes(m)))
1335     xlabel('rel fc prob')
1336 end
1337 print(gcf,'-dpasc','-append','out.ps')
1338 print(gcf, '-depasc', 'ergebnisse/BS_res vgl normH-normHC.eps');
1339 saveas(gcf,'ergebnisse/BS_res vgl normH-normHC.fig')
1340
1341 close(gcf)
1342
1343
1344 %ROC-Diagramm fuer norm1
1345 fig13_1=figure('Units','normalized','Position',[0 0 1 1]);

```

```

1346 set(fig13_1,'name','ROC-Diagramm für norm1','numbertitle','off')
1347 for m=1:num_dim
1348     subplot(3,2,m)
1349     hold on
1350     line([0 1],[0,1])
1351     plot(POFD1(m,:),fcs),POD1(m,:),fcs),'b*')
1352     title(sprintf('norm1: ROC fuer %s',for_axes(m)))
1353     xlabel('POFD')
1354     ylabel('POD')
1355 end
1356 print(gcf,'-dpasc','-append','out.ps')
1357 close(gcf)
1358
1359
1360 %ROC-Diagramm fuer norm2
1361 fig13_2=figure('Units','normalized','Position',[0 0 1 1]);
1362 set(fig13_2,'name','ROC-Diagramm für norm2','numbertitle','off')
1363 for m=1:num_dim
1364     subplot(3,2,m)
1365     hold on
1366     line([0 1],[0,1])
1367     plot(POFD2(m,:),fcs),POD2(m,:),fcs),'r*')
1368     title(sprintf('norm2: ROC fuer %s',for_axes(m)))
1369     xlabel('POFD')
1370     ylabel('POD')
1371 end
1372 print(gcf,'-dpasc','-append','out.ps')
1373 close(gcf)
1374
1375
1376 %ROC-Diagramm fuer norm3
1377 fig13_3=figure('Units','normalized','Position',[0 0 1 1]);
1378 set(fig13_3,'name','ROC-Diagramm für norm3','numbertitle','off')
1379 for m=1:num_dim
1380     subplot(3,2,m)
1381     hold on
1382     line([0 1],[0,1])
1383     plot(POFD3(m,:),fcs),POD3(m,:),fcs),'g*')
1384     title(sprintf('norm3: ROC fuer %s',for_axes(m)))
1385     xlabel('POFD')

```

```

1386     ylabel('POD')
1387 end
1388 print(gcf, '-dpsc', '-append', 'out.ps')
1389 close(gcf)
1390
1391
1392 %ROC-Diagramm fuer norm1-norm2
1393 fig13_4=figure('Units','normalized','Position',[0 0 1 1]);
1394 set(fig13_4,'name','ROC-Diagramm für normH und normHC','numbertitle','off')
1395 for m=1:num_dim
1396     subplot(3,2,m)
1397     hold on
1398     line([0 1],[0,1],'Color','k')
1399     plot(POFD1(m,:),fcs),POD1(m,:),fcs),'b*')
1400     plot(POFD2(m,:),fcs),POD2(m,:),fcs),'r*')
1401     title(sprintf('normH und normHC: ROC fuer %s',for_axes(m)))
1402     xlabel('POFD')
1403     ylabel('POD')
1404 end
1405 print(gcf, '-dpsc', '-append', 'out.ps')
1406 print(gcf, '-depesc', 'ergebnisse/ROC vgl normH-normHC.eps');
1407 saveas(gcf, 'ergebnisse/ROC vgl normH-normHC.fig')
1408 close(gcf)
1409
1410
1411 %post script file in pdf konvertieren Johanna
1412 %name_pdf=sprintf('out_for_fc_step_%g.pdf',fc_steps(fcs));
1413 %name_pdf=sprintf('out_for_c1=%g_c2=%g.pdf',c1,c2);
1414 name_pdf='out.pdf';
1415 ps2pdf('psfile', 'out.ps', 'pdffile', name_pdf, ...
1416     'gspapersize', 'a4', 'verbose', 1, ...
1417     'gscommand', 'C:\Program Files\gs\gs9.06\bin\gswin32c.exe', ...
1418     'gsfontpath', 'C:\Program Files\gs\gs9.06\lib', ...
1419     'gslibpath', 'C:\Program Files\gs\gs9.06\lib')
1420
1421 % %post script file in pdf konvertieren Leo
1422 % ps2pdf('psfile', 'out.ps', 'pdffile', 'name_pdf', ...
1423 %     'gspapersize', 'a4', 'verbose', 1, ...
1424 %     'gscommand', 'C:\Program Files (x86)\gs\gs9.06\bin\gswin32c.exe', ...
1425 %     'gsfontpath', 'C:\Program Files (x86)\gs\gs9.06\lib', ...

```

```

1426 %      'gslibpath', 'C:\Program Files (x86)\gs\gs9.06\lib')
1427
1428 end
1429
1430 end
1431
1432 end
1433
1434
1435 %%
1436 %-----Functions-----
1437
1438 function forcing=forcing(~,X)
1439
1440     varf_3=a*pi*(b*f*pi*X(6)+b)/e;
1441
1442     forcing=[0 varf_1*X(3) 0 0 varf_2 0; ...
1443             -varf_1*X(3) 0 0 -varf_2 0 0; ...
1444             0 0 0 0 0 0; ...
1445             0 -varf_3 varf_4*X(5) 0 0 0; ...
1446             varf_3 0 -varf_4*X(4) 0 0 0;...
1447             -varf_5*X(5) varf_5*X(4) 0 0 0 0]*X;
1448
1449 end
1450
1451
1452 function jacobi=jacobi(X)
1453
1454     var_1=a*(a^2-3)*pi^2/(2*b*(1+a^2));
1455     var_2=a*e*r*sig/(b*pi*(1+a^2));
1456     var_3=a*pi*(b*f*pi*X(6)+b)/e;
1457     var_4=a*pi^2/(2*b);
1458     var_5=a*b*f*pi^2/e;
1459     var_6=a*b*e*pi^2/(2*f);
1460
1461     jacobi=[0 var_1*X(3) var_1*X(2) 0 var_2 0; ...
1462            -var_1*X(3) 0 -var_1*X(1) -var_2 0 0; ...
1463            0 0 0 0 0 0; ...
1464            0 -var_3 var_4*X(5) 0 var_4*X(3) -var_5*X(2); ...
1465            var_3 0 -var_4*X(4) -var_4*X(3) 0 var_5*X(1); ...

```

```

1466         -var_6*X(5) var_6*X(4) 0 var_6*X(2) -var_6*X(1) 0];
1467
1468     end
1469
1470
1471     function H_norm=H_norm(X)
1472         H_norm=zeros(num_dim);
1473         H_norm(1,1)=sqrt((1+a^2)*b^4*pi^3);
1474         H_norm(2,2)=H_norm(1,1);
1475         H_norm(3,3)=sqrt(2*pi^3);
1476         H_norm(6,6)=sqrt(4*r*b^2*f*sig/X(6));
1477         H_norm=1/(4*a*b^2*pi).*H_norm;
1478     end
1479
1480
1481     function C_norm=C_norm(X) %nicht quadratisch
1482         C_norm=zeros(num_dim);
1483         varx=(1+a^2)*b^2*e*pi;
1484         C_norm(1,1)=sqrt(varx*X(4)/X(1));
1485         C_norm(2,2)=sqrt(varx*X(5)/X(2));
1486         C_norm(3,3)=sqrt(4/X(3));
1487         C_norm(6,6)=sqrt(4*f*pi*X(3)/X(6));
1488         C_norm=sqrt(pi/(2*a*b))*C_norm;
1489     end
1490
1491
1492     function S_norm=S_norm(X)
1493         S_norm=zeros(num_dim);
1494         S_norm(3,3)=sqrt(12*f*X(6)/X(3)^2);
1495         S_norm(4,4)=sqrt(3*e^2*pi);
1496         S_norm(5,5)=S_norm(4,4);
1497         S_norm(6,6)=sqrt(6*f^2*pi);
1498         S_norm=r*sig/(12*a*pi)*S_norm;
1499     end
1500
1501
1502     function norm1=norm1(X)
1503         norm1=H_norm(X);
1504     end
1505

```

```

1506
1507 function norm2=norm2(X)
1508     norm2=H_norm(X)+c1.*C_norm(X);
1509 end
1510
1511
1512 function norm3=norm3(X)
1513     norm3=H_norm(X)+c1.*C_norm(X)+c2.*S_norm(X);
1514     %norm3=H_norm(X)+c1.*S_norm(X);
1515 end
1516
1517
1518 function angle=angle(X1,X2) % Winkel zwischen 2 Vektoren in Grad
1519     angle=acos((X1'*X2)/(norm(X1)*norm(X2)))*180/pi;
1520 end
1521
1522
1523 function H_norm_cons_check=H_norm_cons_check(X)
1524     H_norm_cons_check=1/(4*a*b^2*pi)*((1+a^2)*b^4*pi^3*(X(1)^2+X(2)^2)...
1525         +2*pi^3*X(3)^2+4*r*b^2*f*sig*X(6));
1526 end
1527
1528
1529 function C_norm_cons_check=C_norm_cons_check(X)
1530     C_norm_cons_check=(-pi/(2*a*b)*((1+a^2)*b^2*e*pi*(X(1)*X(4)+X(2)*X(5))
1531         ...
1532         +4*f*pi*X(3)*X(6)+4*X(3)));
1533 end
1534
1535 function S_norm_cons_check=S_norm_cons_check(X)
1536     S_norm_cons_check=r*sig/(12*a*pi)*(3*e^2*pi*(X(4)^2+X(5)^2)...
1537         +6*f^2*pi*X(6)^2+12*f*X(6));
1538 end
1539
1540 close all
1541
1542 end

```

**rbc\_6dim\_autom.m**

```

1 function rbc_6dim_autom(lload,randgen)
2
3 close all
4
5 dt=0.001;
6 t_end=20;
7 dt_ana=0.1;
8 n=30;
9 c1=-1.1;
10 c2=0;
11 emp_fac=1;
12
13 einc=1;
14
15 if(lload)
16     load('bsarray')
17 else
18
19     if (randgen)
20
21         t_end_std=100; %30
22         dt_ana_std=0.1;
23         n_std=50;
24         xostd=randn(6,t_end_std/dt_ana_std);
25
26         Alphastd=random('norm',0,1,2,n_std);
27         Alphastd(Alphastd>3)=3;
28         Alphastd(Alphastd<-3)=-3;
29
30         %save('randarray')
31     else
32
33         load('randarray')
34     end
35
36
37 xachse=c1; %für verschiedene c1
38
39 BS_norm1=zeros(6,length(c1),length(c2));

```



```

40 BS_norm2=zeros(6,length(c1),length(c2));
41 BS_norm3=zeros(6,length(c1),length(c2));
42
43 BS_rel_norm1=zeros(6,length(c1),length(c2));
44 BS_rel_norm2=zeros(6,length(c1),length(c2));
45 BS_rel_norm3=zeros(6,length(c1),length(c2));
46
47 BS_res_norm1=zeros(6,length(c1),length(c2));
48 BS_res_norm2=zeros(6,length(c1),length(c2));
49 BS_res_norm3=zeros(6,length(c1),length(c2));
50
51 BS_norm1_AbisF=zeros(1,length(c1),length(c2));
52 BS_norm2_AbisF=zeros(1,length(c1),length(c2));
53 BS_norm3_AbisF=zeros(1,length(c1),length(c2));
54
55 BS_rel_norm1_AbisF=zeros(1,length(c1),length(c2));
56 BS_rel_norm2_AbisF=zeros(1,length(c1),length(c2));
57 BS_rel_norm3_AbisF=zeros(1,length(c1),length(c2));
58
59 BS_res_norm1_AbisF=zeros(1,length(c1),length(c2));
60 BS_res_norm2_AbisF=zeros(1,length(c1),length(c2));
61 BS_res_norm3_AbisF=zeros(1,length(c1),length(c2));
62
63 l=0;
64 for j=1:length(c1)
65     for k=1:length(c2)
66         l=l+1;
67         sprintf('Lauf Nr. %d',l)
68         [ BS_Mat, BS_rel_Mat, BS_res_Mat ]=rbc_6dim(dt,t_end,dt_ana,n,...
69                                                     c1(j),c2(k),emp_fac,xostd,
70                                                     Alphastd);
71         %txt-file mit Brier Scores
72         movefile('out.txt',sprintf('ergebnisse/out_c1=%g,c2=%g.txt',c1(j),c2(k)
73                                     ))
74         %pdf mit Plots der rbc
75         movefile('out.pdf',sprintf('ergebnisse/out_c1=%g,c2=%g.pdf',c1(j),c2(k)
76                                     ))
77         %alle Variablen der rbc-Berechnung
78         movefile('rbc_solution.mat',sprintf('ergebnisse/rbc-solution_c1=%g,c2=%
79                                     g.mat',c1(j),c2(k)))

```

```

76
77     BS_norm1(:,j,k)=BS_Mat(1,:);
78     BS_norm2(:,j,k)=BS_Mat(2,:);
79     BS_norm3(:,j,k)=BS_Mat(3,:);
80
81     BS_rel_norm1(:,j,k)=BS_rel_Mat(1,:);
82     BS_rel_norm2(:,j,k)=BS_rel_Mat(2,:);
83     BS_rel_norm3(:,j,k)=BS_rel_Mat(3,:);
84
85     BS_res_norm1(:,j,k)=BS_res_Mat(1,:);
86     BS_res_norm2(:,j,k)=BS_res_Mat(2,:);
87     BS_res_norm3(:,j,k)=BS_res_Mat(3,:);
88
89     BS_norm1_AbisF(1,j,k)=norm(BS_norm1(:,j,k));
90     BS_norm2_AbisF(1,j,k)=norm(BS_norm2(:,j,k));
91     BS_norm3_AbisF(1,j,k)=norm(BS_norm3(:,j,k));
92
93     BS_rel_norm1_AbisF(1,j,k)=norm(BS_rel_norm1(:,j,k));
94     BS_rel_norm2_AbisF(1,j,k)=norm(BS_rel_norm2(:,j,k));
95     BS_rel_norm3_AbisF(1,j,k)=norm(BS_rel_norm3(:,j,k));
96
97     BS_res_norm1_AbisF(1,j,k)=norm(BS_res_norm1(:,j,k));
98     BS_res_norm2_AbisF(1,j,k)=norm(BS_res_norm2(:,j,k));
99     BS_res_norm3_AbisF(1,j,k)=norm(BS_res_norm3(:,j,k));
100
101     end
102 end
103
104 save('bsarray')
105
106 if (einc)
107     return
108 end
109
110
111 %%
112 %Plots
113
114 for_title=['A' 'B' 'C' 'D' 'E' 'F'];
115

```

```

116 %Brier Score
117 [cc1,cc2]=meshgrid(c1,c2);
118 l1=length(c1);
119 l2=length(c2);
120 BS_norms(:,:,1)=BS_norm1;
121 BS_norms(:,:,2)=BS_norm2;
122 BS_norms(:,:,3)=BS_norm3;
123
124 for in=3:3      %in=1 plottet norm1
125     figure('Units','normalized','Position',[0 0 1 1]);
126     for m=1:6
127         subplot(3,2,m)
128         hold on
129         % plot(c2,reshape(BS_norms(m,2,:,:),[l2,l1]))
130         % plot(c2,reshape(BS_norms(m,2,:,in),[l2,l1]),'r')
131         % plot(c2,reshape(BS_norms(m,2,:,in),[l2,l1]),'g')
132         contour(cc1,cc2,reshape(BS_norms(m,:,:,in),[l2,l1]))
133         % contour(cc1,cc2,reshape(BS_norm2(m,:,:,in),[l2,l1]),'r')
134         % contour(cc1,cc2,reshape(BS_norm3(m,:,:,in),[l2,l1]),'g')
135
136         colorbar
137         title(sprintf('Brier Score für Komponente %s',for_title(m)))
138         xlabel('c_1')
139         ylabel('c_2')
140     end
141 end
142 print(gcf,'-dpsc','-append','BS_out.ps')
143 close(gcf)
144
145 %Brier Score: Norm über alle Komponenten
146 [cc1,cc2]=meshgrid(c1,c2);
147 l1=length(c1);
148 l2=length(c2);
149 BS_norms_AbisF(:,:,1)=BS_norm1_AbisF; %erste Dimension weggeben
150 BS_norms_AbisF(:,:,2)=BS_norm2_AbisF;
151 BS_norms_AbisF(:,:,3)=BS_norm3_AbisF;
152
153
154 for in=3:3      %in=1 plottet norm1
155     figure('Units','normalized','Position',[0 0 1 1]);

```

```

156     hold on
157 %   plot(c2,reshape(BS_norms(m,2,:,:),[12,11]))
158 %   plot(c2,reshape(BS_norms(m,2,:,in),[12,1]),'r')
159 %   plot(c2,reshape(BS_norms(m,2,:,in),[12,1]),'g')
160     contour(cc1,cc2,reshape(BS_norms_AbisF(1,:,:in),[12,11]))
161 %   contour(cc1,cc2,reshape(BS_norm2(m,:,:in),[12,11]),'r')
162 %   contour(cc1,cc2,reshape(BS_norm3(m,:,:in),[12,11]),'g')
163
164     colorbar
165     title('Brier Score für Norm über alle Komponenten')
166     xlabel('c_1')
167     ylabel('c_2')
168 end
169 print(gcf,'-dpsc','-append','BS_out.ps')
170 close(gcf)
171
172
173 %Brier Score: norm1 vs. norm2 vs. norm3
174 figure('Units','normalized','Position',[0 0 1 1]);
175 for m=1:6
176     subplot(3,2,m)
177     hold on
178     plot(xachse(:),BS_norm1(m,:,1),'b')
179     plot(xachse(:),BS_norm2(m,:,1),'r')
180     plot(xachse(:),BS_norm3(m,:,1),'g')
181     legend('norm1','norm2','norm3')
182     title(sprintf('Brier Score für Komponente %s',for_title(m)))
183     xlabel('c_1')
184     ylabel('BS')
185 end
186 print(gcf,'-dpsc','-append','BS_out.ps')
187 close(gcf)
188
189 %Brier Score reliability: norm1 vs. norm2 vs. norm3
190 figure('Units','normalized','Position',[0 0 1 1]);
191 for m=1:6
192     subplot(3,2,m)
193     hold on
194     plot(xachse(:),BS_rel_norm1(m,:,1),'b')
195     plot(xachse(:),BS_rel_norm2(m,:,1),'r')

```

```

196     plot(xachse(:),BS_rel_norm3(m,:,1),'g')
197     legend('norm1','norm2','norm2')
198     title(sprintf('Brier Score reliability für Komponente %s',for_title(m)))
199     xlabel('c_1')
200     ylabel('BS_r_e_l')
201 end
202 print(gcf,'-dpasc','-append','BS_out.ps')
203 close(gcf)
204
205 %Brier Score resolution: norm1 vs. norm2 vs. norm3
206 figure('Units','normalized','Position',[0 0 1 1]);
207 for m=1:6
208     subplot(3,2,m)
209     hold on
210     plot(xachse(:),BS_res_norm1(m,:,1),'b')
211     plot(xachse(:),BS_res_norm2(m,:,1),'r')
212     plot(xachse(:),BS_res_norm3(m,:,1),'g')
213     legend('norm1','norm2','norm3')
214     title(sprintf('Brier Score resolution für Komponente %s',for_title(m)))
215     xlabel('c_1')
216     ylabel('BS_r_e_s')
217 end
218 print(gcf,'-dpasc','-append','BS_out.ps')
219 close(gcf)
220
221 %Brier Score Norm über alle Komponenten: norm1 vs. norm2
222 figure('Units','normalized','Position',[0 0 1 1]);
223 subplot(3,1,1)
224 hold on
225 plot(xachse(:),BS_norm1_AbisF(1,:,1),'b')
226 plot(xachse(:),BS_norm2_AbisF(1,:,1),'r')
227 plot(xachse(:),BS_norm3_AbisF(1,:,1),'g')
228 title('BS - Norm über alle Komponenten')
229 legend('norm1','norm2','norm2')
230 xlabel('c_1')
231 ylabel('BS')
232
233 subplot(3,1,2)
234 hold on
235 plot(xachse(:),BS_rel_norm1_AbisF(1,:,1),'b')

```

```

236 plot(xachse(:),BS_rel_norm2_AbisF(1,:,1),'r')
237 plot(xachse(:),BS_rel_norm3_AbisF(1,:,1),'g')
238 title('BS_r_e_l - Norm über alle Komponenten')
239 legend('norm1','norm2','norm2')
240 xlabel('c_1')
241 ylabel('BS_r_e_l')
242
243 subplot(3,1,3)
244 hold on
245 plot(xachse(:),BS_res_norm1_AbisF(1,:,1),'b')
246 plot(xachse(:),BS_res_norm2_AbisF(1,:,1),'r')
247 plot(xachse(:),BS_res_norm3_AbisF(1,:,1),'g')
248 title('BS_r_e_s - Norm über alle Komponenten')
249 legend('norm1','norm2','norm2')
250 xlabel('c_1')
251 ylabel('BS_r_e_s')
252
253 print(gcf,'-dpsc','-append','BS_out.ps')
254 close(gcf)
255
256 %post script file in pdf konvertieren Johanna
257 ps2pdf('psfile','BS_out.ps','pdffile','BS_out.pdf', ...
258       'gspapersize','a4','verbose',1, ...
259       'gscommand','C:\Program Files\gs\gs9.06\bin\gswin32c.exe', ...
260       'gsfontpath','C:\Program Files\gs\gs9.06\lib', ...
261       'gslibpath','C:\Program Files\gs\gs9.06\lib')
262
263 % %post script file in pdf konvertieren Leo
264 % ps2pdf('psfile','BS_out.ps','pdffile','BS_out.pdf', ...
265 %       'gspapersize','a4','verbose',1, ...
266 %       'gscommand','C:\Program Files (x86)\gs\gs9.06\bin\gswin32c.exe', ...
267 %       'gsfontpath','C:\Program Files (x86)\gs\gs9.06\lib', ...
268 %       'gslibpath','C:\Program Files (x86)\gs\gs9.06\lib')
269
270 movefile('bsarray.mat','ergebnisse/bsarray.mat')
271 movefile('BS_out.pdf','ergebnisse/BS_out.pdf')
272 delete('BS_out.ps')
273
274 end

```

**rbc\_6d\_BSplot\_1.m**

```

1 function rbc_6d_BSplot_1
2 %BS-Plots für c1
3 close all
4
5 vgl=2; %3 Vergleich von 2 oder 3 versch. Normen
6 closefig=1; %0: fig offen lassen, 1:fig schließen
7
8 load('BSarray_Lauf1')
9 load('BSarray_Lauf2')
10
11 %für Lauf1/Lauf2:
12     BS_normH=BS_norm1;
13     BS_normHC=BS_norm2;
14     BS_normHS=BS_norm3;
15
16     BS_rel_normH=BS_rel_norm1;
17     BS_rel_normHC=BS_rel_norm2;
18     BS_rel_normHS=BS_rel_norm3;
19
20     BS_res_normH=BS_res_norm1;
21     BS_res_normHC=BS_res_norm2;
22     BS_res_normHS=BS_res_norm3;
23
24     BS_normH_AbisF=BS_norm1_AbisF;
25     BS_normHC_AbisF=BS_norm2_AbisF;
26     BS_normHS_AbisF=BS_norm3_AbisF;
27
28     BS_rel_normH_AbisF=BS_rel_norm1_AbisF;
29     BS_rel_normHC_AbisF=BS_rel_norm2_AbisF;
30     BS_rel_normHS_AbisF=BS_rel_norm3_AbisF;
31
32     BS_res_normH_AbisF=BS_res_norm1_AbisF;
33     BS_res_normHC_AbisF=BS_res_norm2_AbisF;
34     BS_res_normHS_AbisF=BS_res_norm3_AbisF;
35
36
37 %%
38 %Plots
39

```

```

40 for_title=['A' 'B' 'C' 'D' 'E' 'F'];
41 colordef white
42
43 %Brier Score: H-Norm vs. HC-Norm (vs. HS-Norm)
44 figure('Units','normalized','Position',[0 0 1 1]);
45 for m=1:6
46     subplot(3,2,m)
47     hold on
48     plot(xachse(:),BS_normH(m,:,1),'b') %#ok<NODEF> -> aus load
49     plot(xachse(:),BS_normHC(m,:,1),'r')
50     %legend('H-Norm','HC-Norm')
51     if vgl==3;
52         plot(xachse(:),BS_normHS(m,:,1),'g')
53         %legend('H-Norm','HC-Norm','HS-Norm')
54     end
55     title(sprintf('Brier Score für Komponente %s',for_title(m)))
56     xlabel('c_1')
57     ylabel('BS')
58 end
59 print(gcf,'-dpsc','-append','BS_plot.ps')
60 print(gcf, '-depesc', 'ergebnisse/BS_komp.eps');
61 saveas(gcf,'ergebnisse/BS_komp.fig')
62 if (closefig)
63     close(gcf)
64 end
65
66 %Brier Score reliability: H-Norm vs. HC-Norm (vs. HS-Norm)
67 figure('Units','normalized','Position',[0 0 1 1]);
68 for m=1:6
69     subplot(3,2,m)
70     hold on
71     plot(xachse(:),BS_rel_normH(m,:,1),'b')
72     plot(xachse(:),BS_rel_normHC(m,:,1),'r')
73     %legend('H-Norm','HC-Norm')
74     if vgl==3;
75         plot(xachse(:),BS_rel_normHS(m,:,1),'g')
76         %legend('H-Norm','HC-Norm','HS-Norm')
77     end
78     title(sprintf('Brier Score reliability für Komponente %s',for_title(m)))
79     xlabel('c_1')

```



```

80     ylabel('BS_r_e_1')
81
82     if m==3 || m==6
83         ylim([-1 1])
84     else
85         set(gca, 'YTickLabel', num2str(get(gca,'YTick'),'','%0.3f'))
86     end
87 end
88 print(gcf, '-dpsc', '-append', 'BS_plot.ps')
89 print(gcf, '-depsc', 'ergebnisse/BS_rel_komp.eps');
90 saveas(gcf, 'ergebnisse/BS_rel_komp.fig')
91 if (closefig)
92     close(gcf)
93 end
94
95 %Brier Score resolution: H-Norm vs. HC-Norm (vs. HS-Norm)
96 figure('Units','normalized','Position',[0 0 1 1]);
97 for m=1:6
98     subplot(3,2,m)
99     hold on
100    plot(xachse(:),BS_res_normH(m,:,1),'b')
101    plot(xachse(:),BS_res_normHC(m,:,1),'r')
102    %legend('H-Norm','HC-Norm')
103    if vgl==3;
104        plot(xachse(:),BS_res_normHS(m,:,1),'g')
105        %legend('H-Norm','HC-Norm','HS-Norm')
106    end
107    title(sprintf('Brier Score resolution für Komponente %s',for_title(m)))
108    xlabel('c_1')
109    ylabel('BS_r_e_s')
110 end
111 print(gcf, '-dpsc', '-append', 'BS_plot.ps')
112 print(gcf, '-depsc', 'ergebnisse/BS_res_komp.eps');
113 saveas(gcf, 'ergebnisse/BS_res_komp.fig')
114 if (closefig)
115     close(gcf)
116 end
117
118 %Brier Score Norm über alle Komponenten: H-Norm (vs. HC-Norm) nur BS
119 figure('Units','normalized','Position',[0 0 1 1]);

```

```

120 hold on
121 plot(xachse(:),BS_normH_AbisF(1,:,1),'b')
122 plot(xachse(:),BS_normHC_AbisF(1,:,1),'r')
123 legend('H-Norm','HC-Norm')
124 if vgl==3;
125     plot(xachse(:),BS_normHS_AbisF(1,:,1),'g')
126 end
127 title('BS - Norm über alle Komponenten')
128 legend('H-Norm','HC-Norm','HS-Norm')
129 xlabel('c_1')
130 ylabel('BS')
131
132 print(gcf,'-dpsc','-append','BS_plot.ps')
133 print(gcf,'-depsc','ergebnisse/BS.eps');
134 saveas(gcf,'ergebnisse/BS.fig')
135 if (closefig)
136     close(gcf)
137 end
138
139 %Brier Score Norm über alle Komponenten: H-Norm vs. HC-Norm (vs. HC-Norm)
140 figure('Units','normalized','Position',[0 0 1 1]);
141 subplot(3,1,1)
142 hold on
143 plot(xachse(:),BS_normH_AbisF(1,:,1),'b')
144 plot(xachse(:),BS_normHC_AbisF(1,:,1),'r')
145 %legend('H-Norm','HC-Norm')
146 if vgl==3;
147     plot(xachse(:),BS_normHS_AbisF(1,:,1),'g')
148 end
149     title('BS - Norm über alle Komponenten')
150 %legend('H-Norm','HC-Norm','HS-Norm')
151 xlabel('c_1')
152 ylabel('BS')
153
154 subplot(3,1,2)
155 hold on
156 plot(xachse(:),BS_rel_normH_AbisF(1,:,1),'b')
157 plot(xachse(:),BS_rel_normHC_AbisF(1,:,1),'r')
158 %legend('H-Norm','HC-Norm')
159 if vgl==3;

```

```

160     plot(xachse(:),BS_rel_normHS_AbisF(1,:,1),'g')
161 end
162     title('BS_r_e_l - Norm über alle Komponenten')
163 %legend('H-Norm','HC-Norm','HS-Norm')
164 xlabel('c_1')
165 ylabel('BS_r_e_l')
166
167 subplot(3,1,3)
168 hold on
169 plot(xachse(:),BS_res_normH_AbisF(1,:,1),'b')
170 plot(xachse(:),BS_res_normHC_AbisF(1,:,1),'r')
171 %legend('H-Norm','HC-Norm')
172 if vgl==3;
173     plot(xachse(:),BS_res_normHS_AbisF(1,:,1),'g')
174 end
175 title('BS_r_e_s - Norm über alle Komponenten')
176 %legend('H-Norm','HC-Norm','HS-Norm')
177 xlabel('c_1')
178 ylabel('BS_r_e_s')
179
180 print(gcf,'-dpsc','-append','BS_plot.ps')
181 print(gcf, '-depnc', 'ergebnisse/BS_BSrel_BSres.eps');
182 saveas(gcf,'ergebnisse/BS_BSrel_BSres.fig')
183 if (closefig)
184     close(gcf)
185 end
186
187 %post script file in pdf konvertieren Johanna
188 ps2pdf('psfile', 'BS_plot.ps', 'pdffile', 'BS_plot.pdf', ...
189     'gspapersize', 'a4', 'verbose', 1, ...
190     'gscommand', 'C:\Program Files\gs\gs9.06\bin\gswin32c.exe', ...
191     'gsfontpath', 'C:\Program Files\gs\gs9.06\lib', ...
192     'gslibpath', 'C:\Program Files\gs\gs9.06\lib')
193
194 movefile('BS_plot.pdf','ergebnisse/BS_plot.pdf')
195 delete('BS_plot.ps')
196
197
198 end

```

**rbc\_6d\_BSplot\_2.m**

```

1 function rbc_6d_BSplot_2
2 %BS-Plots für c1/c2 (Contourplots)
3 close all
4
5 closefig=1; %0: fig offen lassen, 1:fig schließen
6
7 load('BSarray_Lauf3')
8
9 %für Lauf3
10     BS_normH=BS_norm1;
11     BS_normHC=BS_norm2;
12     BS_normHCS=BS_norm3;
13
14     BS_normH_AbisF=BS_norm1_AbisF;
15     BS_normHC_AbisF=BS_norm2_AbisF;
16     BS_normHCS_AbisF=BS_norm3_AbisF;
17
18 %%
19 %Plots
20
21 for_title=['A' 'B' 'C' 'D' 'E' 'F'];
22
23 %Brier Score
24 [cc1,cc2]=meshgrid(c1,c2);
25 l1=length(c1);
26 l2=length(c2);
27 BS_norms(:,:,1)=BS_normH;
28 BS_norms(:,:,2)=BS_normHC;
29 BS_norms(:,:,3)=BS_normHCS;
30
31 %Brier Score: HCS-Norm komponentenweise
32 for in=3:3 %in=3 plottet norm3 -> also Lk aus H-,C-, und S-Norm
33     figure('Units','normalized','Position',[0 0 1 1]);
34     for m=1:6
35         subplot(3,2,m)
36         hold on
37         contour(cc2,cc1,reshape(BS_norms(m,:,:),in),[l2,l1])
38         colorbar
39         title(sprintf('Brier Score für Komponente %s',for_title(m)))

```

```

40     xlabel('c_1') %c1 und c2 vertauscht weil Matrix sonst transponiert
        werden müsste
41     ylabel('c_2')
42     end
43 end
44 print(gcf, '-dpsc', '-append', 'BS_plot.ps')
45 print(gcf, '-depsc', 'ergebnisse/BS_komp.eps');
46 saveas(gcf, 'ergebnisse/BS_komp.fig')
47 if (closefig)
48     close(gcf)
49 end
50
51 %Brier Score: HCS-Norm über alle Komponenten
52 [cc1,cc2]=meshgrid(c1,c2);
53 l1=length(c1);
54 l2=length(c2);
55 BS_norms_AbisF(:,:,:,1)=BS_normH_AbisF; %erste Dimension weggeben
56 BS_norms_AbisF(:,:,:,2)=BS_normHC_AbisF;
57 BS_norms_AbisF(:,:,:,3)=BS_normHCS_AbisF;
58
59
60 for in=3:3 %in=1 plottet norm1
61     figure('Units','normalized','Position',[0 0 1 1]);
62     hold on
63     contour(cc2,cc1,reshape(BS_norms_AbisF(1,:,:,:),[l2,l1]))
64     colorbar
65     title('Brier Score für Norm über alle Komponenten')
66     xlabel('c_1') %c1 und c2 vertauscht weil Matrix sonst transponiert werden
        müsste
67     ylabel('c_2')
68 end
69 print(gcf, '-dpsc', '-append', 'BS_plot.ps')
70 print(gcf, '-depsc', 'ergebnisse/BS.eps');
71 saveas(gcf, 'ergebnisse/BS.fig')
72 if (closefig)
73     close(gcf)
74 end
75
76 %post script file in pdf konvertieren Johanna
77 ps2pdf('psfile', 'BS_plot.ps', 'pdffile', 'BS_plot.pdf', ...

```

```
78     'gspapersize', 'a4', 'verbose', 1, ...
79     'gscommand', 'C:\Program Files\gs\gs9.06\bin\gswin32c.exe', ...
80     'gsfontpath', 'C:\Program Files\gs\gs9.06\lib', ...
81     'gslibpath', 'C:\Program Files\gs\gs9.06\lib')
82
83 movefile('BS_plot.pdf', 'ergebnisse/BS_plot.pdf')
84 delete('BS_plot.ps')
85
86
87 end
```

# Abbildungsverzeichnis

2.1	Versuchsaufbau Rayleigh-Bénard-Konvektion . . . . .	6
2.2	Hexagonale Konvektionsstrukturen und Konvektionsrollen . . . . .	7
2.3	Satellitenbild Südatlantik . . . . .	8
2.4	Versuchsaufbau zur Herleitung des 3-dimensionalen Lorenz(1963)-Modells	15
2.5	Lorenz(1963)-Attraktor im 3-dimensionalen Phasenraum . . . . .	22
2.6	Lorenz(1963)-Attraktor in 2-dimensionalen Schnittebenen . . . . .	22
2.7	Lorenz(1963)-Attraktor: Störungsdivergenz . . . . .	26
2.8	Lorenz(1963)-Attraktor: Sensitivität bezüglich Störungen im Anfangszu- stand . . . . .	27
2.9	Lorenz(1963)-Attraktor: konservativ und dissipativ . . . . .	29
3.1	Geometrische Interpretation des Lorenz(1963)-Attraktors als Schnittmen- ge der beiden Erhaltungsgrößen $H$ und $C$ . . . . .	45
5.1	Darstellung der Programmstruktur . . . . .	61
5.2	Verschiedene Störungsnormen am Lorenz(1963)-Attraktor . . . . .	74
5.3	Zeitliche Entwicklung der Erhaltungsgrößen bei imaginären Gewichtungsmatrizen . . . . .	81
5.4	Zeitliche Entwicklung der Erhaltungsgrößen bei reellen Gewichtungsmatrizen . . . . .	83
6.1	2d Phasenraumschnitte, $t_{\text{end}}=5$ . . . . .	96
6.2	3d Phasenraumschnitte, $t_{\text{end}}=5$ . . . . .	97
6.3	2d Phasenraumschnitte, $t_{\text{end}}=10$ . . . . .	99
6.4	3d Phasenraumschnitte, $t_{\text{end}}=10$ . . . . .	100
6.5	Übersicht: Zeitliche Entwicklung der Komponenten $A - F$ . . . . .	102
6.6	Zeitliche Entwicklung der Komponente $A$ . . . . .	103
6.7	Zeitliche Entwicklung der Komponente $A$ - Zoom . . . . .	104
6.8	Strömungs- und Temperaturfeld $t = 1.4$ . . . . .	105
6.9	Strömungs- und Temperaturfeld $t = 1.8$ . . . . .	105
6.10	Freie Entwicklung der Komponenten $A - F$ . . . . .	106

6.11 Zeitliche Entwicklung der Erhaltungsgrößen . . . . .	107
6.12 Freie Vorhersage - Vorhersage mit Updateschritten, $dt\_ana=0.5$ . . . . .	108
6.13 Freie Vorhersage - Vorhersage mit Updateschritten, $dt\_ana=0.1$ . . . . .	109
6.14 Freie Vorhersage - Vorhersage mit Updateschritten, $dt\_ana=0.5$ . . . . .	110
6.15 Spur der Fehlerkovarianzmatrix $P$ vor dem Update . . . . .	112
6.16 Korrelation rms, spread und Spur( $P$ ) . . . . .	113
6.17 BS der gesamten Prognose, $c1=[-5,5]$ . . . . .	115
6.18 BS der gesamten Prognose, Aufspaltung in Zuverlässigkeit und Auflösung, $c1=[-5,5]$ . . . . .	116
6.19 BS komponentenweise, $c1=[-5,5]$ . . . . .	117
6.20 BS reliability, komponentenweise, $c1=[-5,5]$ . . . . .	118
6.21 BS resolution, komponentenweise, $c1=[-5,5]$ . . . . .	119
6.22 Brier Scores für $c=-1$ im Vergleich zur Energienorm, komponentenweise .	119
6.23 Brier Scores für $c1=-1$ , Norm über alle Komponenten im Vergleich zur Energienorm . . . . .	120
6.24 BS der gesamten Prognose, $c1=[-2,0]$ . . . . .	121
6.25 BS der gesamten Prognose, Aufspaltung in Zuverlässigkeit und Auflösung, $c1=[-2,0]$ . . . . .	122
6.26 BS komponentenweise, $c1=[-2,0]$ . . . . .	123
6.27 BS reliability, komponentenweise, $c1=[-2,0]$ . . . . .	124
6.28 BS resolution, komponentenweise, $c1=[-2,0]$ . . . . .	125
6.29 Brier Scores für $c=-1.1$ im Vergleich zur Energienorm, komponentenweise	125
6.30 Verbesserungen der physikalischen Feldprognosen für $c1=-1.1$ im Vergleich zur Energienorm . . . . .	126
6.31 Brier Scores für $c1=-1.1$ , Norm über alle Komponenten im Vergleich zur Energienorm . . . . .	127
6.32 Reliability-Diagramm . . . . .	128
6.33 Histogramm der vorhergesagten Wahrscheinlichkeitsklassen . . . . .	129
6.34 ROC-Diagramm, H-Norm und HC-Norm . . . . .	130
6.35 BS der gesamten Prognose, $c1=[-5,5]$ . . . . .	131
6.36 BS der gesamten Prognose, Aufspaltung in Zuverlässigkeit und Auflösung, $c1=[-5,5]$ . . . . .	132
6.37 BS komponentenweise, $c1=[-5,5]$ . . . . .	133
6.38 BS reliability, komponentenweise, $c1=[-5,5]$ . . . . .	134
6.39 BS resolution, komponentenweise, $c1=[-5,5]$ . . . . .	135
6.40 Contourplot komponentenweise, $c1=[-5,5]$ , $c2=[-5,5]$ . . . . .	136
6.41 Contourplot, Norm über alle Komponenten, $c1=[-5,5]$ , $c2=[-5,5]$ . . . . .	137



## *Abbildungsverzeichnis*

7.1	BS der gesamten Prognose . . . . .	138
7.2	Verbesserung der Brier Scores in Zuverlässigkeit und Auflösung gegenüber der Energienorm für $c_1=-1.1$ . . . . .	139
7.3	Verbesserungen der physikalischen Feldprognosen gegenüber der Energie- norm für $c_1=-1.1$ . . . . .	139
7.4	Contourplot, Norm über alle Komponenten, $c_1=[-5,5]$ , $c_2=[-5,5]$ . . . . .	140

# Tabellenverzeichnis

3.1	Vergleich: Geostrophischer Wind - Kanonische Dynamik . . . . .	39
5.1	Bedeutung der Modellparameter . . . . .	63
5.2	Fehlerkovarianzmatrizen . . . . .	70
5.3	Größen zur Verifikation . . . . .	90
5.4	Kontingenztafel . . . . .	91

# Literaturverzeichnis

- [1] Arnol'd, V. I.: Mathematical methods of classical mechanics, Springer Verlag, 1989
- [2] Bihlo A.: A tutorial on Hamiltonian mechanics, Uni Wien, 2011
- [3] Bihlo A.: Ausgewählte Themen der dynamischen Meteorologie, Uni Wien, 2011
- [4] Bihlo A.: Rayleigh-Bénard convection as a Nambu-metrisplectic problem, Journal of Physics A: Mathematical and Theoretical, Vol.41, 2008
- [5] Bihlo A.: Symmetry methods in the atmospheric sciences, Dissertation, Uni Wien, 2010
- [6] Bihlo A., Popovych R.O.: Symmetry justification of Lorenz' maximum simplification, Nonlinear Dynamics, Vol. 61, Issue 1-2 , S. 101-107, 2010
- [7] Bihlo A., Staufer J.: Minimal atmospheric finite-mode models preserving symmetry and generalized Hamiltonian structures, Elsevier, Physica D, S.599 - 606, 2010
- [8] Chen Z.-M., Price W. G.: On the relation between Rayleigh-Bénard convection and Lorenz system, Elsevier, Chaos, Solitons and Fractals 28, S. 571 - 578, 2006
- [9] Dankert J.: Die ode-Functions in Matlab, [http://www.rzbt.haw-hamburg.de/dankert/WWWErgVert/html/matlab\\_\\_die\\_ode-functions\\_\\_muss.html](http://www.rzbt.haw-hamburg.de/dankert/WWWErgVert/html/matlab__die_ode-functions__muss.html), 12.04.2012
- [10] Darryl D. H.: Applications of Poisson Geometry to Physical Problems, arXiv („The Archive”), Mathematics > Dynamical Systems, 2007
- [11] European Centre for Medium-Range Weather Forecasts, [http://www.ecmwf.int/newsevents/training/rcourse\\_notes/PREDICTABILITY/PREDICTING\\_UNCERTAINTY/Predicting\\_uncertainty7.html](http://www.ecmwf.int/newsevents/training/rcourse_notes/PREDICTABILITY/PREDICTING_UNCERTAINTY/Predicting_uncertainty7.html), 14.09.2012
- [12] European Centre for Medium-Range Weather Forecasts, <http://www.ecmwf.int/research/ifsd docs/CY25r1/Ensemble/Ensemble-3-2.html>, 21.11.2012
- [13] Elert G.: Measuring Chaos, Kapitel 4.3: Lyapunov Exponent, 2007, <http://hypertextbook.com/chaos/43.shtml>, 23.10.2012

- [14] Errico et al.: The spectra of singular values in a regional model, *Tellus*, Vol. 53, 2003
- [15] Fäßler A.: Mütter H., Wharam D. : Analytische Mechanik, Kapitel2: Hamilton Formalismus, Kanonische Transformationen, Universität Tübingen, 2005, <http://www.uni-tuebingen.de//uni/ptf1/>, 26.06.2012
- [16] Haimberger L.: Vorhersagbarkeit und Ensemblevorhersage, Uni Wien, 2010
- [17] Haimberger L.: Numerische Methoden der Wettervorhersage, Uni Wien, 2005
- [18] Jönck U., Prill F.: Das Lorenz-System - Seminar über gewöhnliche Differentialgleichungen, Universität Hamburg - Fachbereich Mathematik, 2003, <http://www.math.uni-hamburg.de/home/lauterbach/scripts/seminar03/prill.pdf>, 14.07.2012
- [19] Jung A.: Nicht lineare Dynamik und Chaos - Lorenz-System und seltsame Attraktoren, <http://andreas.welcomes-you.com/research/talks/lorenz>, 28.06.2012
- [20] Kleeman L.: Understanding and Applying Kalman Filtering, Department of Electrical and Computer Systems Engineering Monash University, Clayton, <http://www.ecse.monash.edu.au/centres/irrc/LKPPubs/Kalman.PDF>, 12.10.2012
- [21] Lange H.J.: Die Physik des Wetters und des Klimas, Kapitel 10: Die Energie-Wirbel-Theorie, S. 529 - 559, Reimer, 2002
- [22] Lewis et al.: Dynamic Data Assimilation: A Least Squares Approach, Cambridge University Press, 2006
- [23] Lorenz E.N.: Deterministic Nonperiodic Flow, *Journal of the Atmospheric Sciences*, Vol.20, S. 130 - 141, 1963
- [24] Manneville P.: Dynamics of Spatio-Temporal Cellular Structures: Rayleigh-Bénard convection, thirty years of experimental, theoretical, and modeling work, S. 41 - 65, Springer Verlag, 2006,
- [25] Morrison P.J., Hazeltine R.D.: Hamiltonian formulation of reduced magnetohydrodynamics, *Physics of Fluids*, Vol.27, 1984
- [26] Nevir, P.: Die Nambu-Felddarstellung der Hydro-Thermodynamik und ihre Bedeutung für die dynamische Meteorologie, Habilitationsschrift, 1998
- [27] Névir P.: Energie-Wirbel-Theorie der atmosphärischen Dynamik, Uni Wien, 2006
- [28] Névir P.: Blender R.: Hamiltonian and Nambu Representation of the Non-Dissipative Lorenz Equations, *Beitr. Phys. Atmosph.*, Vol. 67, S.133 - 140, 1994
- [29] Nolting W.: Grundkurs Theoretische Physik 2, Springer Verlag, 8. Aufl., 2010
- [30] Olver P. J.: Application of Lie Groups to Differential Equations, Springer Verlag, 2000

- [31] Rothman D.H.: Nonlinear Dynamics I: Chaos, Massachusetts Institute of Technology, <http://ocw.mit.edu/courses/earth-atmospheric-and-planetary-sciences/12-006j-nonlinear-dynamics-i-chaos-fall-2006/lecture-notes/course.pdf>, 25.06.2012
- [32] Saltzman B.: Finite Amplitude Free Convection as an Initial Value Problem I, Journal of Atmospheric Sciences, vol. 19, Issue 4, S. 329-341, 1962
- [33] Scheck, F.: Theoretische Physik 1: Klassische Mechanik, Springer Verlag, 2007
- [34] Schlichting H.J.: Backhaus U., Küpker H.G.: Chaos beim Wasserrad – Ein einfaches mechanisches Modell für das Lorenz-System, Universität Duisburg Essen, 1991, [http://www.didaktik.physik.uni-due.de/veranstaltungen/SS2011/Chaos/Chaos\\_beim\\_Wasserrad.pdf](http://www.didaktik.physik.uni-due.de/veranstaltungen/SS2011/Chaos/Chaos_beim_Wasserrad.pdf), 13.07.2012
- [35] Schlichting H.: Gersten K.: Grenzschicht-Theorie, Springer Verlag, 10. Auflage, 2006
- [36] Schweizer A.: [http://www.schweizer-fn.de/stoff/begriffe/v2\\_begriffe.htm](http://www.schweizer-fn.de/stoff/begriffe/v2_begriffe.htm), 30.6.2012
- [37] Siefert M., Renner C.: Rayleigh-Bénard-Konvektion, Universität Oldenburg - Fachbereich Physik, <http://twist.physik.uni-oldenburg.de/anleitung.pdf>, 23.06.2012
- [38] Simon D., Seminar Fraktale, Kapitel 13: Dynamische Systeme, Universität Ulm, 2006, [http://www.mathematik.uni-ulm.de/stochastik/lehre/ws06\\_07/seminar\\_fraktale/ausarbeitung\\_simon.pdf](http://www.mathematik.uni-ulm.de/stochastik/lehre/ws06_07/seminar_fraktale/ausarbeitung_simon.pdf), 24.06.2012
- [39] Sommer M.: Numerische und geometrische Aspekte der Energie-Wirbel-Theorie, Dissertation, Freie Universität Berlin, 2010
- [40] Tessmann I.: Zum Lorenz Modell deterministischer, nichtperiodischer Strömung, Technische Universität Hamburg-Harburg, 1990, <http://www.tuhh.de/rzt/rzt/it/lorenz/lorenz.pdf>, 23.06.2012
- [41] Thies R.: Chaos - Nichtlineare Dynamik, Universität Dortmund - Fachbereich Informatik, 2004, <http://ls11-www.cs.uni-dortmund.de/lehre/sas/Chaostheorie.pdf>, 04.08.2012

# Curriculum Vitae

## Persönliche Daten

Name: Johanna Maria Bugkel  
Geburtsdatum: 28.07.1989  
Geburtsort: Baden / Nö

## Ausbildung

1999 - 2007 BG und BRG Baden Frauengasse  
seit 2007 Studium der Meteorologie, Wien

## Berufserfahrung

Juli 2005, 2006, 2007 Sportbetreuung Leistungszentrum  
Juli 2008, 2009, 2010 Apotheke Zum Schutzengel  
2005 - 2009 Wettkampfleiterin im Nö Judo Landesverband  
2008 Ausbildung BSO Sportmanagerin  
April - Sept. 2012 Zentralanstalt für Meteorologie und Geodynamik:  
Verarbeitung von Radionuklidmessungen  
des International Monitoring Systems  
der CTBTO  
seit 2005 Judotrainerin  
seit 2011 Sozialbetreuerin Obdachlosenzentrum Gruft 2

## Weitere Kenntnisse

Sprachen: Deutsch, Englisch, Spanisch, Latein  
EDV-Kenntnisse: Windows, Linux, Shell Scripts, Matlab, LaTeX,  
MS Office, Fotobearbeitung

# Danksagung

Bedanken möchte ich mich bei meinem Betreuer Ao. Univ.-Prof. Mag. Dr. Leopold Haimberger nicht nur insgesamt für die kompetente und motivierte Unterstützung der Arbeit, sondern besonders auch für konkrete Hilfestellungen in der programmiertechnischen Realisierung des untersuchten Modells.

Weiters gilt mein Dank auch Dr. Alexander Bihlo, von ihm stammt das verwendete Differentialgleichungssystem, er war bei dessen Bearbeitung und Erforschung stets sehr hilfsbereit und nahm sich wie auch mein Betreuer selbst immer Zeit für meine Vorschläge und Fragen.

Natürlich gebührt mein Dank ebenfalls meinen Eltern Hedwig und Erich Bugkel für das Ermöglichen meiner Ausbildung und auch meinem Freund Daniel, der mich durch mein gesamtes Studium mit viel Geduld und Verständnis begleitet hat, möchte ich an dieser Stelle herzlichst für alles danken.